

Харківський національний університет радіоелектроніки

Факультет Комп'ютерні науки

Кафедра Системотехніки

Освітньо-кваліфікаційний рівень другий (магістерський)

Спеціальність 6.050101 – Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри

(підпис)

« ___ » _____ 20 __ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ (ПРОЕКТ)

студентові Бугаю Данилу Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Дослідження застосування алгоритмів
коллаборативної фільтрації в туристичній установі

затверджена наказом по університету від « ___ » _____ 20 __ р. № _____

2. Термін подання студентом роботи (проекту) 26 травня 2020

3. Вихідні дані до роботи (проекту) вимоги до веб-системи, що буде розроблена,
середовище проектування IntelliJ IDEA, мова розробки Java, сервер Apache Tomcat,
середовище проектування Toad for MySQL, декларативна мова програмування SQL,
сервер MySQL, гнучкий клієнт (Google Chrome або Firefox).

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити) вступ,
аналіз проблемної галузі та моделювання предметної області, аналіз проблемної
галузі та конкурентів, опис та аналіз структурних і функціональних особливостей
об'єкту дослідження, аналіз рекомендаційних алгоритмів, методи на основі контенту,
методи коллаборативної фільтрації, гібридні методи дослідження механізму
коллаборативної фільтрації, опис прийнятих програмних рішень, моделювання
даних, проектування програмного забезпечення.

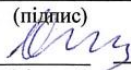
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів) 5.1 Контекстна діаграма (1 аркуш формату А4), 5.2 Діаграма потоків даних (1 аркуш формату А4), 5.3 Схема рекомендацій за колаборативної фільтрації (1 аркуш формату А4), 5.4 Діаграма варіантів використання (1 аркуш формату А4), 5.5 Модель даних (1 аркуш формату А4), 5.6 Алгоритм роботи пошуку (1 аркуш формату А4), 5.7 Діаграма послідовності (1 аркуш формату А4), 5.8 Діаграма кооперації (1 аркуш формату А4), 5.9 Діаграма станів (1 аркуш формату А4), 5.10 Модель предметної області (1 аркуш формату А4).

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	<i>Отримання завдання на атестаційне проектування</i>	<i>30.03.2020</i>	
2	<i>Аналіз завдання, пошук літератури та аналогів з теми атестаційної роботи</i>	<i>07.04.2020</i>	
3	<i>Опрацювання літератури та аналіз об'єкту дослідження</i>	<i>15.04.2020</i>	
4	<i>Розробка моделі взаємодії даних</i>	<i>21.04.2020</i>	
5	<i>Розробка структури зберігання даних</i>	<i>01.05.2020</i>	
6	<i>Написання коду програмної системи</i>	<i>07.05.2020</i>	
7	<i>Аналіз результатів, отриманих за допомогою програмного засобу</i>	<i>12.05.2020</i>	
8	<i>Оформлення пояснювальної записки та програмної документації</i>	<i>15.05.2020</i>	
9	<i>Оформлення графічної частини та презентаційних матеріалів комп'ютерного</i>	<i>18.05.2020</i>	
10	<i>Представлення на рецензування</i>	<i>20.05.2020</i>	

Студент Бугай Д. О.

Керівник роботи (проекту)  проф., д. т. н. Мінухін С.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить: 73 с., 41 рис.

Мета роботи – підвищення ефективності роботи туристичної установи шляхом використання рекомендаційних алгоритмів та інтегрування певного алгоритму в інформаційну систему туристичної установи.

Предмет дослідження – рекомендаційні алгоритми, вбудовані в інформаційну систему туристичної установи, що призначена для пошуку і бронювання готелів для користувача на основі його історії бронювання і кореляції з іншими користувачами.

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.

У роботі був проведений аналіз заданої предметної області та процесів, які в ній протікають, проведений огляд існуючих рекомендаційних алгоритмів та їх порівняльний аналіз, що дозволив визначити який з них найкраще підійде до даної інформаційної системи.

РЕКОМЕНДАЦІЯ, АЛГОРИТМ, МЕТОДОЛОГІЯ, ФІЛЬТРАЦІЯ,
КОЛЛАБОРАЦІЯ, СХОЖІСТЬ, РІШЕННЯ.

ABSTRACT

Explanatory note to the performance appraisal contains: 73 s., 41 figs.

The subject of the study is the analysis of the implementation of the recommendation algorithms in the information system for the tourist institution that deals with the search and reservation of hotels in order to select the most appropriate hotel for the user based on its booking history and correlation with other users.

The purpose of the work is to analyze the given subject area and the processes that take place in it, to review the existing recommendation algorithms and to determine which one is best suited to the system under consideration and to integrate the recommendation mechanism into the existing system.

Given the current level of development of information systems, the introduction of recommendation algorithms to existing web systems is very relevant. Recommendation algorithms can save a lot of time for system users and owners to keep customers on the resource and help them make decisions.

In the work the analysis of the given subject area and the processes that take place in it was carried out, the review of the existing recommendation algorithms and their comparative analysis was conducted, which allowed to determine which of them is best suited to this information system.

RECOMMENDATION, ALGORITHM, METHODOLOGY, FILTRATION,
COLLABORATION, SIMILARITY, DECISION.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

SADT (акронім від англ. Structured analysis and design technique) - методологія структурного аналізу і проектування, інтегруюча процес моделювання, управління конфігурацією проекту, використання додаткових засобів мови і керівництво проектом зі своєю графічною мовою.

IDEF - методології сімейства ICAM (Integrated Computer-Aided Manufacturing) для вирішення завдань моделювання складних систем, дозволяє відображати і аналізувати моделі діяльності широкого спектру складних систем в різних розрізах.

IDEF0 - методологія функціонального моделювання (англ. Function modeling) і графічна нотація, призначена для формалізації і опису бізнес-процесів.

IDEF1 (integration definition for information modeling) - одна з методологій сімейства IDEF. Застосовується для побудови інформаційної моделі, яка представляє структуру інформації, необхідної для підтримки функцій виробничої системи або середовища.

CF – колаборативна фільтрація. Це техніка, що використовується в системах рекомендацій.

САПР (CAD або CADD) - система автоматизованого проектування.

DFD - загальноприйняте скорочення від англ. data flow diagrams - діаграми потоків даних. Так називається методологія графічного структурного аналізу, що описує зовнішні по відношенню до системи джерела і адресати даних, логічні функції, потоки даних і сховища даних, до яких здійснюється доступ.

ЗМІСТ

ВСТУП.....	8
1 Аналіз проблемної галузі та моделювання предметної області	11
1.1 Аналіз проблемної галузі та конкурентів.....	11
1.2 Опис та аналіз структурних і функціональних особливостей об'єкту дослідження.....	21
1.3 Висновки за розділом.....	27
2 Аналіз рекомендаційних алгоритмів	29
2.1 Методи на основі контенту	31
2.2 Методи колаборативної фільтрації.	34
2.3 Гібридні методи.	39
2.4 Дослідження механізму колаборативної фільтрації	43
2.5 Висновки за розділом.....	58
3 Опис прийнятих програмних рішень.....	59
3.1 Розробка структури БД.....	61
3.2 Проектування програмного забезпечення	64
3.3 Опис обраних технологій та демонстрація	71
3.4 Висновки за розділом	80
ВИСНОВОК	81
ДОДАТОК А Графічний матеріал атестаційної роботи.....	84
ДОДАТОК Б Текст програми	98
ДОДАТОК В Відомість атестаційної роботи	109

ВСТУП

Враховуючи сучасний рівень розвитку інформаційних технологій використання комп'ютера для збереження будь-яких видів інформації стає єдиним засобом, що надає широкі можливості керування інформацією. Одну з найважливіших ролей у сучасному процесі отримання інформації відіграє мережа Інтернет. Вона з самого початку свого розвитку орієнтована на інформаційне забезпечення своїх користувачів. У зв'язку з цим розвиток технологій електронної комерції на сучасному етапі набуває першочергового значення.

Актуальність впровадження Інтернет магазинів, сайтів які дозволяють он-лайн резервувати квитки на поїзд та літак чи номери в готелі являє собою важливу задачу, адаптовану до потреб різних компаній-виробників і має декілька переваг від звичайних магазинів чи звичайного резервування. По-перше, використання Інтернет технологій дозволяє зменшити витрати на транспорт та час витрачений на пошук потрібного товару чи готелю. Інтернет ресурси, завдяки зручній навігації та дружньому інтерфейсу дозволяють за мінімальних затрат часу ознайомитися з повним набором послуг, які пропонує певна організація. По-друге, наприклад, плануючи подорож мінімізується ймовірність потрапляння в такі незручні ситуації при бронюванні номеру в готелі, як невідповідність тому, що розповіли по телефону тощо.

В той самий час сьогодні все більшої популярності та широкого застосування набувають системи рекомендацій. Вони змінюють спосіб неживих веб-сайтів спілкуватися зі своїми користувачами. Замість того, щоб надавати статичний досвід, у якому користувачі шукають та потенційно купують певні продукти, системи рекомендацій збільшують взаємодію, щоб забезпечити більш багатий досвід користувачів, за рахунок зменшення часу на пошук відповідного товару чи

послуги та тим самим приваблюють все більшу кількість клієнтів що безпосередньо збільшує прибуток власникам систем.

Системи рекомендацій автоматично визначають рекомендації для окремих користувачів на основі минулих покупок і пошуків та поведінки інших користувачів. Компанії, що використовують системи рекомендацій, зосереджуються на збільшенні продажів внаслідок дуже персоналізованих пропозицій та покращеного досвіду клієнтів.

Рекомендації, як правило, прискорюють пошук і полегшують користувачам доступ до товарів, які їх цікавлять, і дивують їх пропозиціями, які вони ніколи не шукали б. Більше того, компанії можуть залучати та утримувати клієнтів, розсилаючи електронні листи із посиланнями на нові пропозиції, що відповідають інтересам одержувачів.

Тож колаборативна фільтрація фільтрує інформацію, використовуючи взаємодії та дані, зібрані системою від інших користувачів. Вона оснований на ідеї, що люди, які погодилися в оцінці певних питань, швидше за все, згодом згодяться. Концепція проста: коли ми хочемо знайти новий фільм для перегляду, ми часто просимо наших друзів поради. Природно, ми більше довіряємо рекомендаціям друзів, які поділяють смаки, подібні до наших.

Більшість систем фільтрації застосовують так звану методику на основі індексу подібності. У підході, що базується на кореляції, кількість користувачів вибирається виходячи зі схожості з активним користувачем. Висновок для активного користувача робиться шляхом обчислення середньозваженого показника оцінок вибраних користувачів. Системи спільної фільтрації фокусуються на взаємозв'язку між користувачами та предметами. Подібність предметів визначається схожістю оцінок цих позицій користувачами, які оцінили обидві позиції.

Підсумовуючи можна зробити висновок що системи рекомендацій надають персоналізовану інформацію, вивчаючи її через історію взаємодії з цим

користувачем. Так само, як алгоритми машинного навчання, система рекомендацій робить передбачення на основі попередньої поведінки користувача. Зокрема, вона розроблена для прогнозування переваг користувачів для набору елементів на основі історії.

Метою даної роботи є створення веб-ресурсу, що містить каталог готелів за певним напрямком та має можливість на основі механізму рекомендацій швидко рекомендувати відповідний готель на основі історії бронювання та кореляції з іншими користувачами, тим самим ефективно зберігаючи його час.

Необхідно розв'язати задачу впровадження рекомендаційного алгоритму в інформаційну систему для туристичної установи, яка займається пошуком і бронюванням готелів задля того щоб здійснювати підбір найбільш відповідного готелю для користувача на основі його історії бронювання і кореляції з іншими користувачами. Це дозволить заощаджувати велику кількість часу користувачам системи а власникам утримувати клієнтів на ресурсі та допомагати їм прийняти рішення в разі того якщо вони матимуть сумніви стосовно кращого вибору.

Для вирішення цієї задачі необхідно провести аналіз існуючих рекомендаційних алгоритмів та виявити який з них найкраще підійде до системи що розглядається. Для того аби це зробити необхідно детально проаналізувати усю систему та виявити критерії на основі яких можна буде будувати рекомендаційний механізм.

Як результат необхідно буде проаналізувати отриману модель та зробити висновки щодо ефективності та успішності роботи рекомендаційного алгоритму в контексті інформаційної системи, що розглядається.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ ТА МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз проблемної галузі та конкурентів

Із поширенням різноманітних електронних засобів та покращенням Інтернет покриття усе більше й більше людей починають користуватися можливостями, які для них відкриває мережа. Завдяки зручності, яка виражається в отриманні повної інформації стосовно усіх послуг готелю та усіх вільних номерів, а також збереженню величезної кількості свого часу велика кількість людей починає віддавати перевагу он-лайн бронюванню номерів готелю, будь то планування відпочинку чи бізнес-поїздка. За даними дослідження пошукових запитів в Google за останній рік кількість запитів типу "бронювати + готель" зросла на 20%. Люди шукають альтернативу пакетним турам, а де простіше забронювати, як не онлайн.

За підсумками цього року сегмент онлайн подорожей - самостійного бронювання подорожей - виріс на 35-40%. З 2010 року він виріс в шість разів. Інтернет-бронювання захопить 25% загального ринку туристичних послуг.

Варто зазначити, що сегмент зростає в першу чергу за рахунок нових клієнтів з офлайна. Це, як правило, люди з не дуже великим інтернет-досвідом, переважно жінки.

Раніше електронним бронюванням користувалися в основному мешканці великих за розмірами міст, де інформаційні технології найбільш розвинуті, а мережеве покриття має достатньо велику швидкість. Зараз, завдяки інтенсивному розвитку технологій та достатньо доступній їх вартості на арену виходять регіони, за рік показавши зростання попиту від 24% до 40%.

Також слід відзначити, що згідно з статистичних даних імовірність покупки туристичних послуг в он-лайні залежить від стажу користування Інтернетом. Зазвичай до третього-п'ятого року "online-плавання" людина дозволяє собі

ризикнути і зробити дорогу покупку в мережі. Інтернет вже щільно огорнув усю Україну та світ, новачків в он-лайні все менше, а значить, у сегмента електронних замовлень подорожей, а як наслідок бронювання готелів онлайн, все більше потенційних клієнтів.

Враховуючі ці тенденції тепер майже у кожного готельного підприємства є власний веб-ресурс, але далеко не кожен сервіс відповідає своїм цілям. Сайт готелю - це повноцінний інструмент продажів і просування на ринку, а не просто красива картинка. Для того, щоб бути ефективним інструментом продажів, ресурс повинен виконувати ряд завдань. По-перше, необхідно підвищувати прямий онлайн-продаж. Враховуючи сучасну, все зростаючу та зростаючу, конкуренцію виплата великих комісійних системам бронювання і просте очікування стають все менш ефективними. Більш того, не всі готелі можуть дозволити собі розкіш платити комісію посередникам в досить великому обсязі. Оптимальною стратегією є пошук інших шляхів підвищення продажів, в першу чергу, використовуючи найбільш доступний з ресурсів - сайт.

По-друге, потрібно утримувати відвідувачів, які вже перейшли на сайт підприємства. Корисний і функціональний контент дозволить помітно збільшити середній час перебування користувача на сторінці. В першу чергу, мова йде про якісні фотографії готелю, які здатні чинити сильний вплив на потенційного клієнта. Також цікавим рішенням буде вбудований модуль путівника, де будуть відображатися цікаві місця поруч з готелем. Подібні речі в комплексі повинні підстобувати інтерес туриста і збільшувати продажі.

Важливим чинником є мотивація клієнта до бронювання. Перш за все власники готелю мають на меті отримати максимум ефективності від свого ресурсу і системи бронювання. Сервіс повинен спонукати потенційних клієнтів забронювати готель. Дана процедура повинна бути простою і зрозумілою, а також підтримуватися формою зворотного зв'язку, оскільки не всі туристи готові відразу замовити номер. Наявність інформації про певні акції та спеціальні пропозиції на

бронювання буде плюсом. Для багатьох інтернет-ресурсів готелю властивий високий ступінь відмов - ситуацій, коли користувач йде з сайту, так і не перейшовши по будь-яким посиланням. Це говорить про те, що контент, дизайн, структура чи інша складова сайту не мотивують користувача до бронювання.

Також слід звертати увагу на зниження вартості залучення гостей в готель. Виплата комісії посередникам значною мірою підвищує витрати на залучення гостя, змушуючи власників готелів втрачати частину виручки. Правильно розроблений сервіс здатний вирішити цю проблему раз і назавжди, дозволяючи власнику готелю зрівнювати обсяги прямих онлайн-продажів з зовнішнім бронюванням, а то і зовсім орієнтуватися виключно на бронювання зі свого ресурсу.

Аналізуючи сьгоднішні пропозиції представлені на ринку бронювання можна виявити цілий ряд проблем, якими характеризуються ті чи інші реалізації. Метою атестаційної роботи буде створення такої системи, що завдяки своїм програмним рішенням нівелює усі знайдені на інших ресурсах недоліки і в той же час пропонує нові шляхи здійснення процесу бронювання.

Один з найпоширеніших недоліків в існуючих системах це занадто довге завантаження контексту. Багато сервісів перевантажують сайти кількістю елементів зі складною анімацією і аудіо-матеріалами, що негативно позначаються на часі завантаження, який повинен бути зведений до мінімуму. Відвідувач просто піде, якщо всі ключові елементи не завантажуться протягом декількох секунд. Добра швидкість завантаження дозволить знизити відсоток відмов потенційних клієнтів і підвищити конверсію відвідувачів в гостей.

Іншим суттєвим недоліком, дуже часто пов'язаним з попереднім, є невиправдана складність інтерфейсу додатку. Багаторівневе меню значно ускладнює процес навігації. Інтуїтивно зрозуміла і звична навігація сприяє тривалому відвідуванню туриста веб-ресурсу. Крім іншого, часто зустрічаються сервіси, де спочатку не зовсім зрозуміло як дістатися до контенту. Основна інформація, необхідна клієнту перед бронюванням зводиться до трьох пунктів:

місце розташування, опис і вартість. Саме на цю інформацію необхідно робити акцент, щоб користувач без проблем міг її знайти.

Ще одним недоліком, який дратує багатьох користувачів і відштовхує їх є наявність на ресурсі зайвих елементів. У прагненні зробити краще, часто роблять поширену помилку по додаванню непотрібних елементів на сайт. Це можуть бути спливаючі вікна при переході на сайт, які ніяк не цікаві користувачеві, зайва анімація, яка збільшує час завантаження сторінки, або музика, яка включається сама по собі і яку неможливо відключити. Подібні елементи значною мірою дратують потенційних клієнтів, формуючи негативний імідж.

При розробці графічного інтерфейсу додатку слід приділити значну кількість уваги безпосередньо самій формі бронювання. Кількість бронювання з ресурсу визначає його ефективність. Перехід до форми бронювання повинен бути помітний і можливий з будь-якої сторінки. При цьому слід виключити багатоступеневу структуру процесу. Введення будь-якої інформації, без якої можна обійтися, повинно бути виключено. Наявність підказок, які супроводжують користувача при бронюванні підвищує ймовірність успішного завершення даного процесу.

Варто приділяти увагу результатам відвідувань, користуючись засобами веб-аналітики. Ключ до підвищення відсотка успішних замовлень лежить у використанні інструментів веб-аналітики, що дозволяють виявити помилки в навігації і інших елементах сайту і своєчасно їх усунути. Для цього необхідно зрозуміти, як сприймає сайт цільова аудиторія, встановивши лічильники (наприклад, Яндекс Метрика або Google Analytics). Статистика покаже рівень інтересу, який проявляють відвідувачі до окремих елементів і сторінок: де вони затримуються на більший час, а звідки відразу йдуть.

Враховуючи сучасну поширеність смартфонів не слід забувати про мобільний додаток. Користувач дивиться сайт не тільки зі стаціонарних комп'ютерів. Темп зростання використання мобільних пристроїв в останні роки значно збільшився. Навіть якщо число замовлень з мобільних пристроїв невелике, користувач

найчастіше обирає готель саме використовуючи смартфон, а бронювання робить вже з персонального комп'ютера. Більш того, позиція готелю в пошукових системах багато в чому залежить від наявності мобільної версії. Коректне відображення сервісу на будь-якому пристрої - не просто гарний тон, а й актуальна бізнес-необхідність.

Також, якщо порівняти сторінки українських і зарубіжних ресурсів на спеціалізованих порталах, стане очевидно, що на українських менше фотографій, і якості вони гіршої. Вітчизняні розробники відносяться до зображень несерйозно. Але ж, згідно зі статистичними дослідженнями, якщо на сторінці завантажена хоча б одна фотографія, то кількість переглядів сторінки збільшується в середньому на 138%. В 2,5 рази можна підвищити ефективність, розмістивши близько 10 фотографій.

Слід зазначити, що тільки розробити інформаційну систему недостатньо, необхідно весь час її супроводжувати і просувати, використовуючи пошукову оптимізацію, контекстну рекламу, e-mail-розсилки, соцмережі та інші інструменти. Необхідно не тільки придумати унікальну торгову пропозицію, але і вміти донести її до аудиторії.

Таким чином, враховуючи сучасний стан розвитку інформаційних технологій та добру поширеність мережі, яка охоплює практично усі, навіть найбільш віддалені населенні пункти, задача створення Інтернет ресурсу по бронювання готелів являє собою достатньо важливий елемент розвитку бізнесу.

Проте, на сьогодні конкуренція на ринку настільки велика що недостатньо просто розробити гарний інтернет ресурс та запустити його. З великою вірогідністю він не буде успішний бо вже існують сотні інших пропозицій і досить складно завоювати своє місце на ринку не маючи певної бази клієнтів, історичного іміджу чи багатомільйонної рекламної кампанії.

Необхідно зробити так аби сервіс вигідно виділявся на фоні конкурентів та пропонував щось таке що неможливо знайти на інших ресурсах та надавав достатній рівень зручно для користувачів.

Велику популярність зараз набирають такі інструменти як механізми рекомендацій. Багато онлайн-компаній використовують системи рекомендацій для посилення взаємодії з користувачами та збагачення торгового потенціалу. Застосування випадків використання рекомендаційних систем швидко розширюється в багатьох аспектах електронної комерції та Інтернет-ЗМІ протягом останніх 4-5 років, і очікується, що ця тенденція збережеться.

Системи рекомендацій можуть потенційно змінити спосіб спілкування веб-сайтів з користувачами та дозволити компаніям максимізувати свою рентабельність інвестицій на основі інформації, яку вони можуть зібрати на основі переваг та покупок кожного клієнта. Вони намагаються зробити рекомендацію щодо товару чи послуги людям. Певним чином алгоритми намагаються звужити вибір для людей, представляючи їм пропозиції, які вони найчастіше купують чи використовують чи які найвірогідніше сподобаються їм на основі історії та кореляції з іншими користувачами. Системи рекомендацій є майже скрізь від Amazon до Netflix, від Facebook до LinkedIn.

Однією з основних потенційних переваг систем рекомендацій є їх здатність постійно проводити калібрування вподобань користувача. Це суттєво допомагає продуктам утримувати клієнтів. Для прикладу, ви набагато рідше переходите на конкурента Netflix, тому що він має чудове відчуття, які фільми та шоу ви можете захотіти переглянути потім. На рисунку 1.1 наведений зручний та в той же час надзвичайно простий інтерфейс Netflix з результатами роботи рекомендаційного механізму. Оскільки більша частина доходів Netflix надходить від фіксованої періодичної підписки, найбільше, рентабельність інвестицій компанії "виграє" з рекомендаційними системами за рахунок утримання користувачів на платформі.

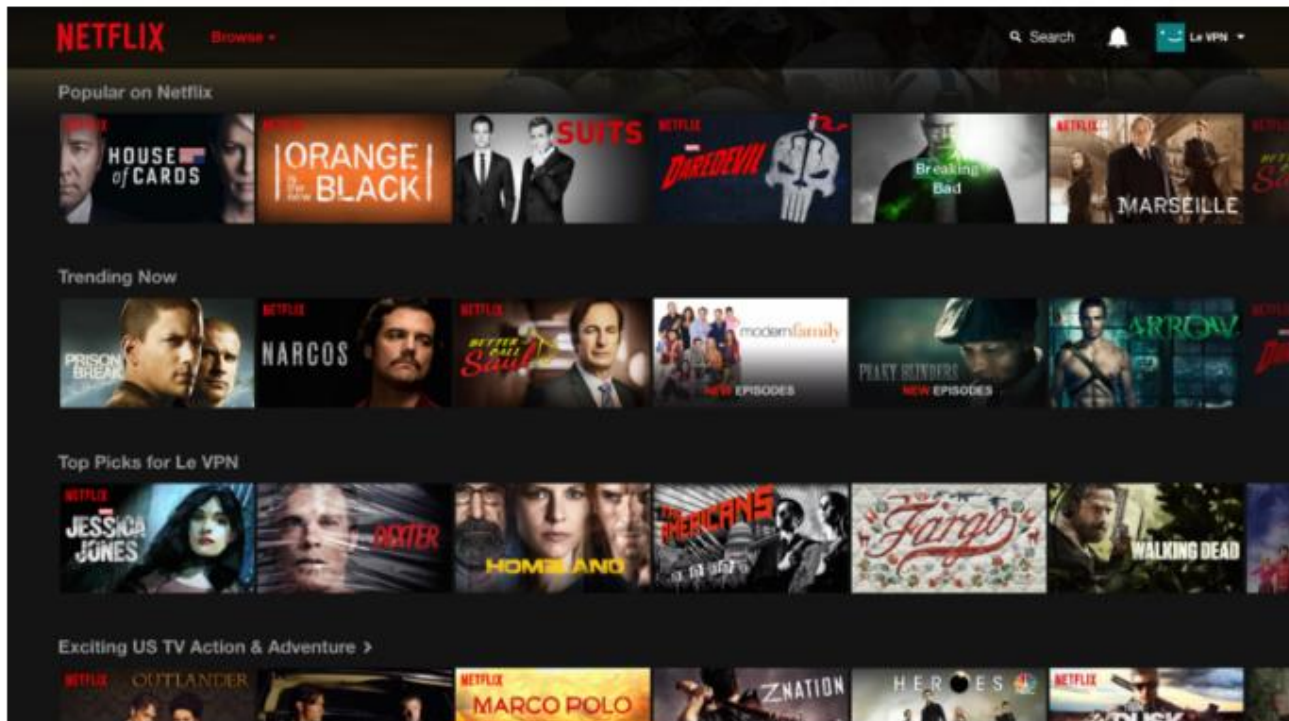
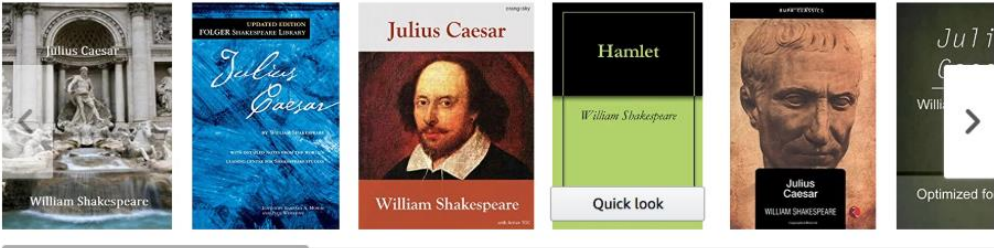


Рисунок 1.1 – Рекомендації на платформі Netflix

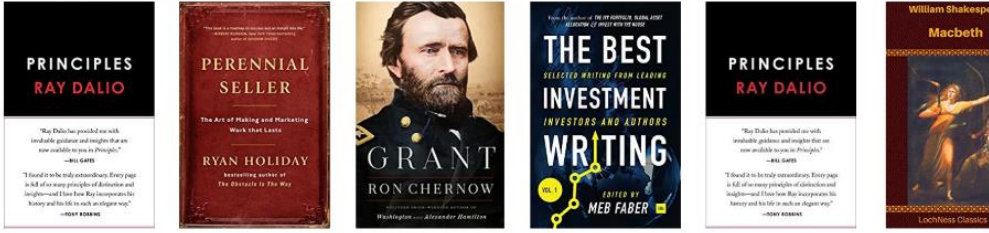
Також варто зазначити що компаніям з великою кількістю товарів буде важко створити певні статичні механізми рекомендацій для кожного окремого користувача, і очевидно, що такі статичні пропозиції швидко застаріють або не матимуть значення для багатьох замовників. Використовуючи різні засоби «фільтрації», гіганти електронної комерції можуть знайти підходящий час для пропонування нових продуктів, які ви, ймовірно, купуватимете (на своєму веб-сайті, електронною поштою чи іншими способами).

Швидка доставка Amazon та прискіпливість при обслуговуванні принесли їм мільйони нових клієнтів. Двигуни рекомендацій відіграють роль не тільки в тому, щоб допомогти клієнтам знайти більше того, що їм потрібно, а також в формуванні кошику таких товарів які буде разом зручніше доставляти, як показано на рисунку 1.2. Тому що якщо компанії не доведеться платити набагато більше за доставку, щоб надіслати вам у два-три рази більше продуктів, їх прибуток покращується.

Inspired by your browsing history [See more](#)



New for you [See more](#)



Customers who bought this item also bought




Рисунок 1.2 – Рекомендації на платформі Amazon

Іноді бачення рентабельності інвестицій не передбачає прямого прохання про оплату. Багато компаній використовують ці системи, щоб просто заохочувати залучення та активність на своєму продукті чи платформі. На YouTube є варіанти передплати, але більшість доходів компанії спрямовуються на рекламу, розміщену в широкому спектрі відео ресурсів. Компанія заробляє більше грошей, коли користувачі знову і знову повертаються. YouTube не оптимізує короткострокову тривалість перегляду. Натомість сервіс має на меті заохотити довгострокове використання, оскільки рекламні перегляди - це рентабельність інвестицій, яку ці системи обслуговують. Facebook - ще один очевидний приклад подібного застосування рекомендаційних двигунів. Важливо також зазначити, що системи

рекомендацій підходять лише компаніям, які мають достатньо даних та талановитих AI, щоб їх добре використовувати, і що для багатьох підприємств та бізнес-моделей краще не використовувати системи рекомендацій, оскільки вони не гарантують, що це кращий підхід, ніж альтернативи.

Попри це, найвідоміші ресурси по бронюванню готелів у світі все ще не використовують рекомендаційні механізми або використовують їх не на повну потужність. Так, якщо ми подивимося на один з найвідоміших у світі ресурсів – booking.com, результати пошуку якого показані на рисунку 1.3 то побачимо що він містить рекомендаційну секцію, проте вона статична.

The screenshot shows the Booking.com interface for a search in Paris. On the left, there is a search filter sidebar with fields for location (Paris), dates, number of adults (2), and children (0). The main content area displays search results for Paris, with a header indicating 4,898 options found. A navigation bar offers filters like 'Our recommendations', 'Hotels and apartments', 'Lowest price first', and 'Stars'. The first result is 'Palym' (3 stars), located in the 12th district of Paris, with a score of 8.6 and 2,630 reviews. A 'Show prices' button is visible next to the listing.

Рисунок 1.3 – Рекомендаційні механізми на booking.com

Не краща ситуація і на такому ресурсі як airbnb. Попри те що сервіс має багато гнучких фільтрів, як показано на рисунку 1.4 все одно він не застосовує рекомендаційні механізми для покращення досвіду користувачів.

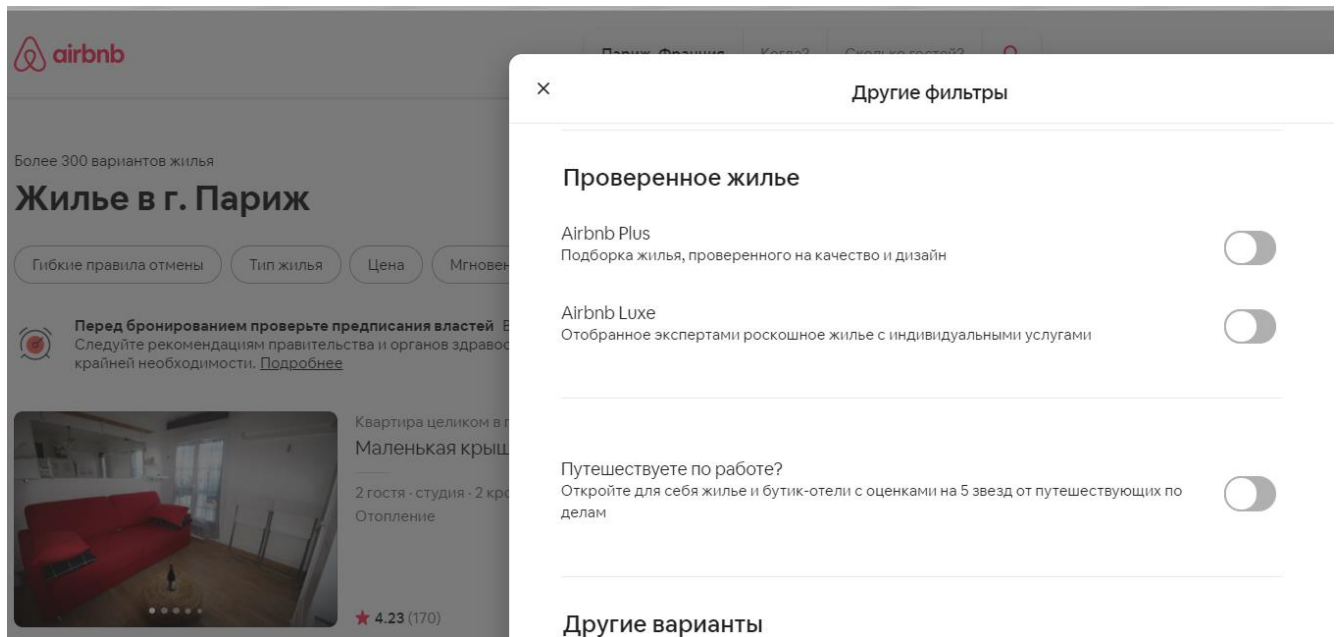


Рисунок 1.4 – Гнучка фільтрація на airbnb

В результаті аналізу існуючих рішень та загального аналізу ринку виявлено що рекомендаційні механізми повсюдно застосовуються в різних інформаційних системах, проте туристичні установи все ще не опанували ці алгоритми та ця область є достатньо перспективною для впровадження рекомендаційних інструментів, що дозволить покращити досвід користувачів тим самим збільшити відсоток успішних бронювань. Певна частина користувачів буде застосовувати рекомендаційні механізми для пришвидшення процесу пошуку готелю і не буде полишати ресурс з метою найти щось швидше та краще на сторонньому сервісі.

1.2 Опис та аналіз структурних і функціональних особливостей об'єкту дослідження

Темпи розвитку сучасного суспільства у всіх його проявах багато в чому визначаються досягнутим рівнем інформатизації. Одним з основних факторів, що впливають на процес інформатизації, є наявність сучасної технології і інструментальних засобів проектування складних інформаційно-керуючих систем, під якими розуміються системи збору, зберігання, обробки інформації незалежно від функціонального призначення. Добре налагоджена технологія є тією основою, на базі якої розрізнені інструментальні технологічні засоби можуть бути об'єднані і представлені у вигляді систем автоматизованого проектування.

Проектування складних ІУС, незалежно від сфери їх застосування, є процесом, що вимагає великих витрат ресурсів і часу, а також залучення великої кількості висококваліфікованих фахівців. Однак при відсутності стандартизованої індустріальної технології проектування немає ніяких гарантій створення ефективної або навіть просто працездатної системи.

Це пояснюється тим, що в даній ситуації якість результатів проектування в основному визначається особистим досвідом, знаннями, евристичними міркуваннями і уподобаннями розробників. В кінцевому рахунку спостерігаються, відсутність наступності при зміні фахівців, труднощі сприйняття і розуміння концепцій розробки через унікальність ІУС.

Альтернативним шляхом є створення високоефективних систем автоматизованого проектування (САПР), що дозволяють в найкоротші терміни проектувати ІУС для будь-якого об'єкта з урахуванням його конкретних особливостей і вимог користувачів [1].

На початкових етапах створення інформаційних систем необхідно зрозуміти, як працює організація, котру збираються автоматизувати. Зробити це достатньо легко, використовуючи методологію SADT. Це одна з найвідоміших методологій

аналізу та проектування систем. Вона є, мабуть, єдиною методологією, що відбиває такі характеристики, як управління, зворотній зв'язок і ресурси. Інша особливість SADT полягає в тому, що вона розвивалася як мова опису функціонування систем загального вигляду, тоді як в інших структурних методологіях упор частіше робиться на проектування програмного забезпечення.

Основна ідея даної методології, власне як і будь-якої іншої структурної методології, полягає в поділі великої задачі на більш маленькі, пов'язані між собою. Кожна задача – певна функція системи. При виділенні функції у неї повинно бути однозначне, чітке ім'я, яке відбиває суть виконуваних дій. Ці функції приймають певні вхідні данні, обробляють їх та видають результат. Такі функції називаються «чорним ящиком». Якщо ящик не видає результат, то в ньому немає сенсу. Основна перевага виділення та використання таких «чорних ящиків» складається в тому, що їх комбінація дозволяє створювати різні системи, а також полегшується процес тестування та ре-конфігурації. Крім того, їх в рази легше розуміти та модифікувати.

В структурних методологіях присутня ієрархічна організація «чорних ящиків». Кожний наступний рівень має більш глибоку деталізацію, ніж попередній, а також використовуються графічні моделі. Варто пам'ятати, що при розробці слід притримуватись принципів структурного аналізу.[2]

Одним з основних принципів є абстрагування. Виділення існуючих на даному етапі аспектів системи і відволікання від неіснуючих, з метою представлення проблеми в більш простому вигляді. Це основоположний принцип структурного моделювання. Також слід дотримуватися принципу формалізації. Під формалізацією слід розуміти чіткий методичний підхід до вирішення проблеми.

В структурному аналізі дуже важливий принцип інкапсуляції, який полягає в інкапсуляції несуттєвих деталей на кожному конкретному етапі, тобто кожна частина знає тільки необхідну їй інформацію і не більше, а також принцип концептуальної спільності. Цей принцип розповідає про те, що необхідно слідкувати одній і тій самій філософії на усіх етапах життєвого циклу.

Методологію IDEF0 можна вважати наступним етапом розвитку добре відомої графічної мови опису функціональних систем SADT. IDEF0 – методологія функціонального моделювання. Вона дозволяє представити нашу систему у вигляді певного набору взаємопов'язаних функцій, блоків. Як правило, моделювання засобами IDEF0 є першим етапом вивчення будь-якої системи.

Але перш ніж приступати до моделювання необхідно визначити границі системи, точку зору та мету моделювання. Межі системи визначають те, кому і відносно чого будуть ставитися запитання. В нашій системі маємо базу даних готелів у власній системі обліку. Безпосередньо розміщенням і обслуговуванням клієнтів займаються співробітники готелів. Необхідно забезпечити надходження замовлення на резервування з веб-ресурсу безпосередньо до системи певного готелю та оповістити клієнта стосовно поточного статусу його замовлення.

Точка зору визначає форму представлення та вибір необхідної інформації. Це та позиція з якої буде будуватися модель. В нашому випадку доречно будувати модель з точки зору адміністратора системи, бо саме він найкраще розуміє і відповідає за ті етапи системи, які будуть автоматизуватись.

Мета моделювання відповідає на запитання що саме ми маємо отримати від моделі і визначає точність моделювання. Мета моделювання також визначає критерії зупинки. По-перше, варто зазначити, що з системою взаємодіють адміністратор ресурсу та клієнти. Клієнти мають змогу переглядати каталог готелів та оформляти резервування. Функція ж адміністратора полягає в тому, що він має змогу переглядати замовлення на резервування і змінювати їх статус. Таким чином ми фактично автоматизуємо процес надання клієнту інформації щодо готелів та номерів, а також можливість онлайн резервування. Мета моделювання – виявлення списку задач покладених на систему. Основні задачі, виявлені в результаті аналізу – надання інформацій щодо готелів і можливість резервування з відслідковуванням статусу замовлення.

На рисунку 1.5 зображена контекстна діаграма, яка відображає основну функцію системи. В якості вхідних даних виступає інформація стосовно клієнтів, яку ми отримаємо після того, як вони заповнять відповідні поля при оформленні замовлення. За управління відповідають інформація щодо готелів, яка надходить з бази даних та службові обов'язки. Механізмами служать програмне забезпечення, без якого автоматизація процесу бронювання була б неможлива, адміністратор, який приймає замовлення та змінює його статус та клієнт. На виході отримаємо гроші.

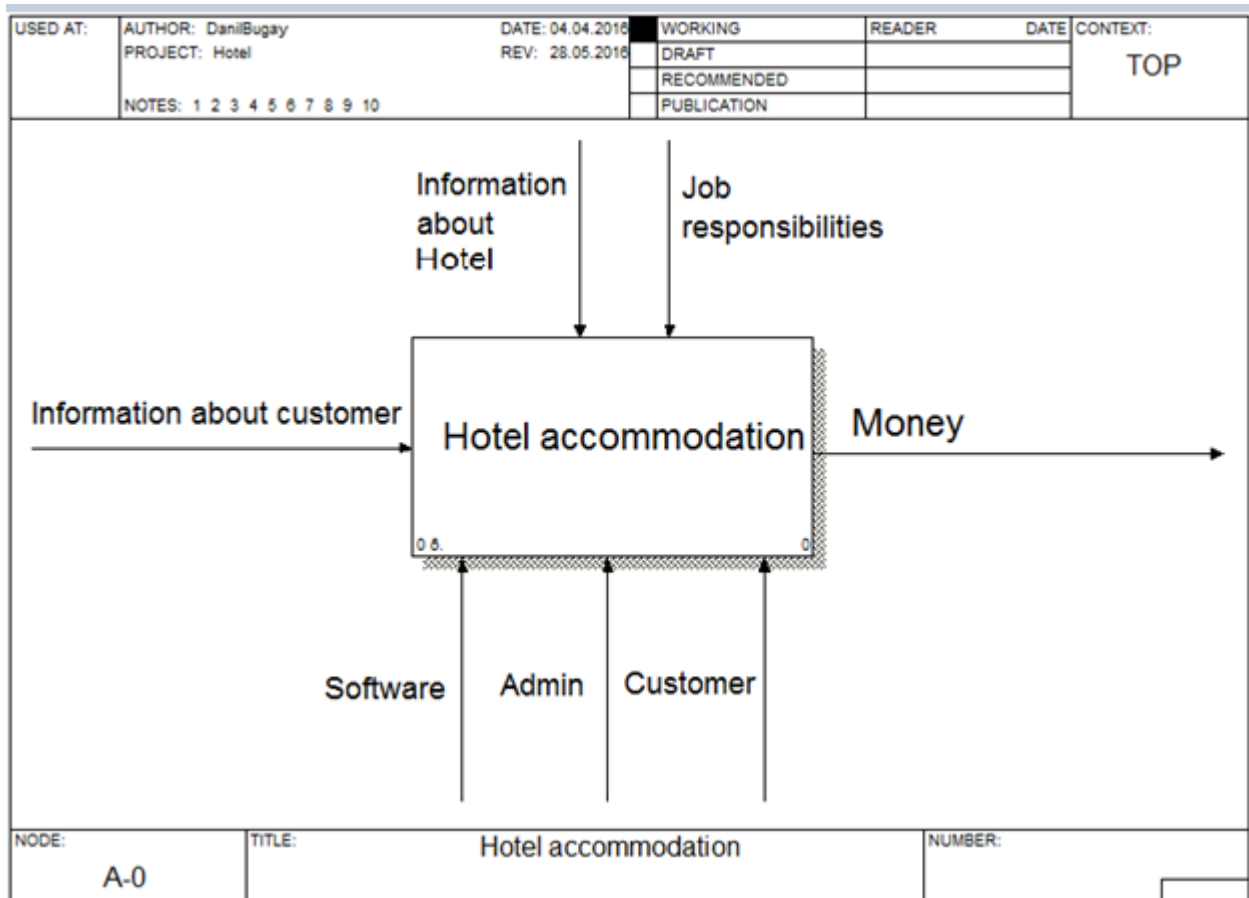


Рисунок 1.5 - Контекстна діаграма предметної області

Задля більш детального проектування широко застосовуються так звані CASE-засоби. Сучасні CASE-засоби охоплюють велику область підтримки

численних технологій проектування ІС: від простих засобів аналізу і документування до повномасштабних засобів автоматизації, що покривають увесь життєвий цикл ПЗ.

Суть моделювання потоків даних при проектуванні складних інформаційно-керуючих систем - це створення так званих діаграм потоків даних. [3]

Методика моделювання потоків даних, забезпечує досить простий спосіб моделювання функціональних систем як на стадії обстеження і аналізу вимог до них, так і на початкових стадіях розробки нових проектних рішень.

В рамках сучасних методологій структурного системного аналізу і проектування, методики моделювання потоків даних використовуються для забезпечення взаєморозуміння з користувачами при ідентифікації вимог до функціонального наповнення ресурсів, як інструмент аналізу в рамках концепції проектування "зверху - вниз" .

На рисунку 1.6 наведено DFD-схему бізнес-процесу "Оформлення бронювання готелю", розробленої в нотації Гейне-Сарсона.[4]

Ця модель дозволяє зрозуміти які потоки даних є в системі, що допоможе при побудові програми при моделюванні сутностей. За допомогою цієї моделі стає зрозуміло які сутності необхідно зберігати в базі даних а які будуть існувати лише на рівні додатку. Згідно з побудованою діаграмою на рівні бази будуть існувати такі елементи як інформація стосовно користувача та інформація стосовно готелів та кімнат.

Для наглядного представлення структури та функціоналу системи необхідно створити моделі за допомогою UML – уніфікованої мови моделювання, що використовується у парадигмі об'єктно-орієнтованого програмування. Різні види діаграм які підтримуються UML і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

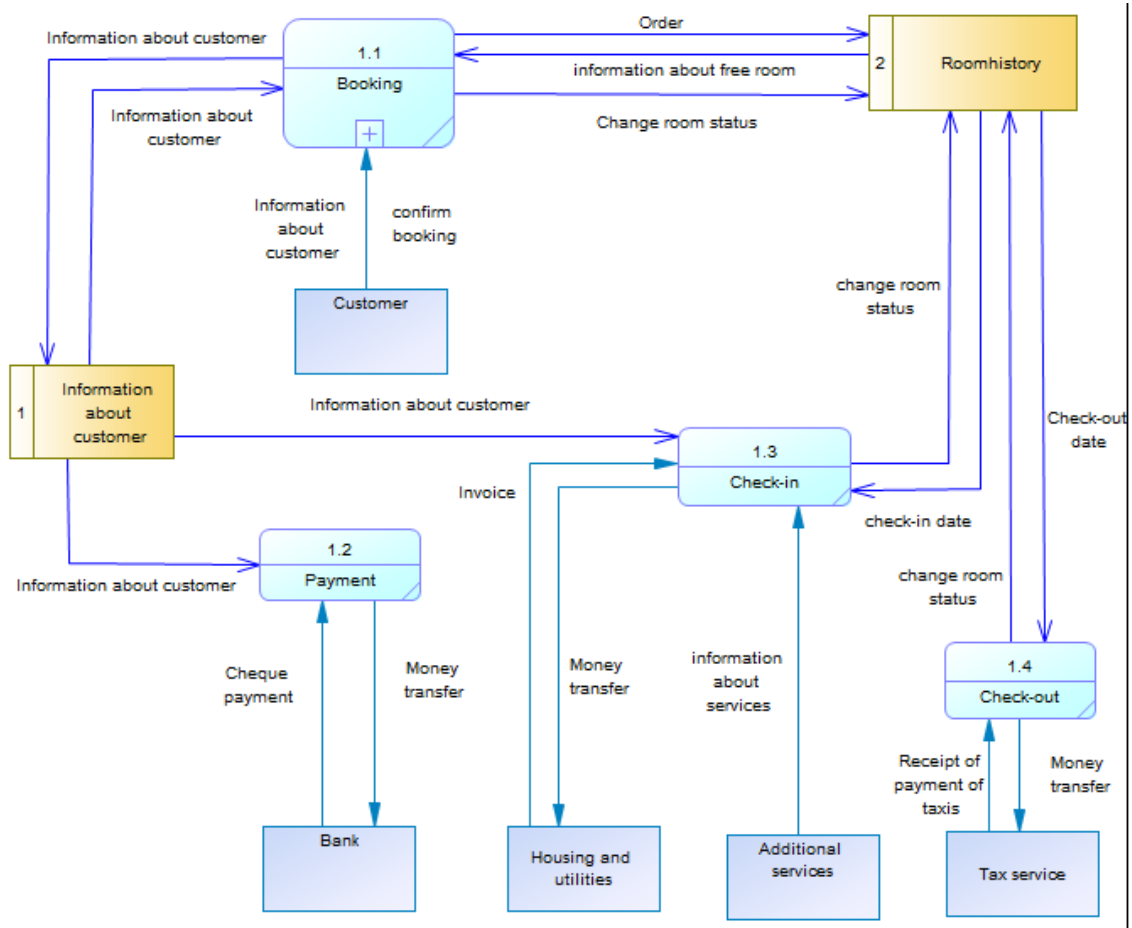


Рисунок 1.6 – Діаграма потоків даних

Загальну інформацію про функціональність системи дає діаграма варіантів використання. Вона побудована за допомогою середовища розробки програмного забезпечення IBM Rational Rose та зображена на рисунку 1.7

Дана діаграма складається з акторів і прецедентів. Актори не є частиною системи - вони являють собою когось чи щось, що має взаємодіяти з системою. У діаграмі були створені актори: Logged users, Unlogged user та Admin, які успадковуються від User, а також Preparation room і Service payment authorization.

За допомогою прецедентів моделюється діалог між актором і системою. Іншими словами, прецеденти визначають можливості, що забезпечуються системою для актора. Набір всіх прецедентів системи визначає способи її використання.

Можна сказати, що прецедент - це послідовність транзакцій, що виконуються системою, яка призводить до значущого результату для певного актора.

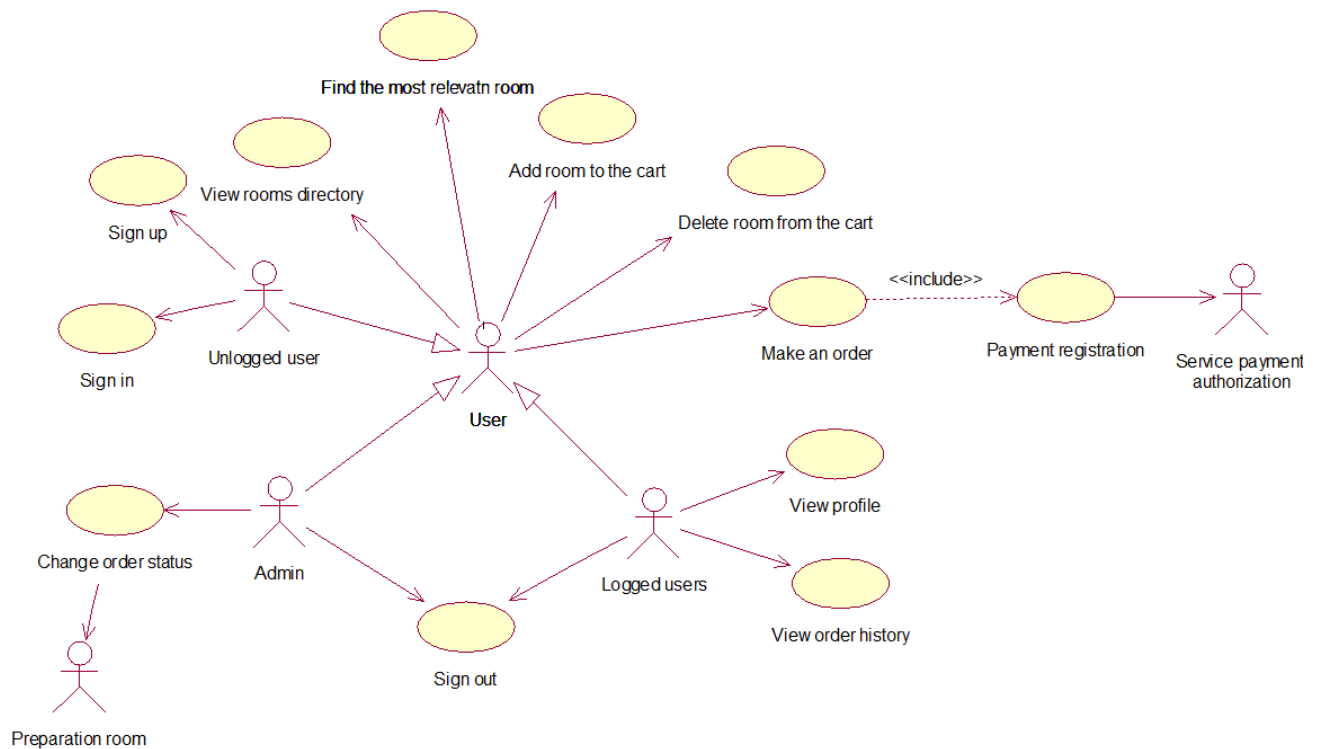


Рисунок 1.7 - Діаграма варіантів використання для туристичної установи

1.3 Висновки за розділом 1

В результаті аналізу існуючих рішень та загального аналізу ринку виявлено що рекомендаційні механізми повсюдно застосовуються в різних інформаційних системах, проте туристичні установи все ще не опанували ці алгоритми та ця область є достатньо перспективною для впровадження рекомендаційних інструментів, що дозволить покращити досвід користувачів тим самим збільшити

відсоток успішних бронювань. Певна частина користувачів буде застосовувати рекомендаційні механізми для пришвидшення процесу пошуку готелю і не буде полишати ресурс з метою знайти щось швидше та краще на сторонньому сервісі.

За результатами дослідження та аналізу предметної області були побудовані ряд діаграм, які допомагають більш детально зрозуміти процеси які відбуваються в туристичній установі та її особливості. Так за допомогою контекстної діаграми було виявлено границі системи та проведене абстрагування. За допомогою діаграми потоків даних було виявлено які сутності слід виділити в системі та які з них будуть зберігатися в базі даних а які будуть існувати на рівні додатку.

Дана система являє собою досить зручний базовий ресурс для масштабування та впровадження рекомендаційних механізмів що не матимуть суттєвого руйнівного ефекту на інші компоненти системи.

Тож в подальшому необхідно розв'язати задачу впровадження рекомендаційного алгоритму в інформаційну систему туристичної установи, яка займається пошуком і бронюванням готелів задля того, щоб здійснювати підбір найбільш відповідного готелю для користувача на основі його історії бронювання і кореляції з іншими користувачами. Це дозволить заощаджувати велику кількість часу користувачам системи а власникам утримувати клієнтів на ресурсі та допомагати їм прийняти рішення в разі того якщо вони матимуть сумніви стосовно кращого вибору.

Для вирішення цієї задачі необхідно провести аналіз існуючих рекомендаційних алгоритмів та виявити який з них найкраще підійде до системи що розглядається, що буде зроблено в наступному розділі. Для цього у розділі була проаналізована система та виявлені аспекти, що найзручніше будуватимуть рекомендаційний механізм на базі кореляції користувача з іншими користувачами.

2 АНАЛІЗ РЕКОМЕНДАЦІЙНИХ АЛГОРИТМІВ

Системи рекомендацій є життєво важливим інструментом, який допомагає подолати проблему перевантаження інформацією. Вони використовуються на більшості веб-сайтів електронної комерції та привертають інтерес широкого наукового співтовариства. Система рекомендацій використовує дані про попередні налаштування користувачів, щоб вибрати нові елементи, які може оцінити окремий користувач.

Вибухонебезпечне зростання кількості доступної цифрової інформації та кількості відвідувачів Інтернету створили потенційну проблему перевантаження інформації, яка перешкоджає своєчасному доступу до цікавих предметів в Інтернеті. Системи пошуку інформації, частково вирішили цю проблему, але пріоритетність та персоналізація (де система відображає наявний зміст на інтереси та вподобання користувача) інформації відсутні. Це збільшило попит на системи рекомендацій більш, ніж будь-коли раніше. Рекомендаційні системи - це системи фільтрації інформації, які вирішують проблему перевантаження інформації, фільтруючи фрагмент життєвої інформації з великої кількості динамічно сформованої інформації відповідно до уподобань, інтересів або спостережуваної поведінки щодо користувача. Система рекомендацій має можливість передбачати, чи певний користувач віддасть перевагу предмету чи ні, на основі профілю користувача [5].

Системи рекомендацій вигідні як постачальникам послуг, так і користувачам. Вони знижують витрати часу на пошук та вибір предметів в онлайн середовищі. Також доведено, що системи рекомендацій покращують процес та якість прийняття рішень. У налаштуваннях електронної комерції системи рекомендацій збільшують дохід, оскільки вони є ефективним засобом продажу більшої кількості товарів. Тому необхідність використання ефективних та точних механізмів рекомендацій у

системі, що дасть відповідні та надійні рекомендації для користувачів, не можна перекреслювати.

В дуже загальному вигляді системи рекомендацій - це алгоритми, спрямовані на те, щоб запропонувати користувачам відповідні предмети (предмети, фільми для перегляду, текст для читання, продукти для придбання або що-небудь інше, залежно від галузей).

Системи рекомендацій - це інструмент, який дозволяє розробникам алгоритмів передбачати, що користувачеві може, а що не може подобатися серед списку заданих елементів. Вони є альтернативою пошуковим полям, оскільки механізми рекомендацій допомагають користувачам виявляти продукти чи інші об'єкти, які вони можуть зовсім не знайти чи не зупинити свою увагу на них.

Системи рекомендацій ідеально працюють одним із двох способів. Вони можуть покладатися на властивості елементів, які користувачеві подобаються. В цьому випадку аналізується історія запитів користувача, щоб визначити, що ще може сподобатися йому а що ні. Або, він може покладатися на симпатії та неприязні інших користувачів, які механізм рекомендацій потім використовує для обчислення індексу подібності між користувачами та рекомендує їм відповідні елементи. Також можна поєднати обидва ці методи для створення набагато більш надійної системи рекомендацій. Однак, як і всі інші проблеми, пов'язані з інформацією, важливо вибрати алгоритм, який підходить для вирішення проблеми найбільше.

Використання ефективних та точних методів рекомендацій є дуже важливим для системи, яка надасть хороші та корисні рекомендації своїм окремим користувачам. Це пояснює важливість розуміння особливостей та потенціалу різних алгоритмів рекомендацій. На рисунку 2.1 наведена класифікація методів рекомендаційних механізмів.

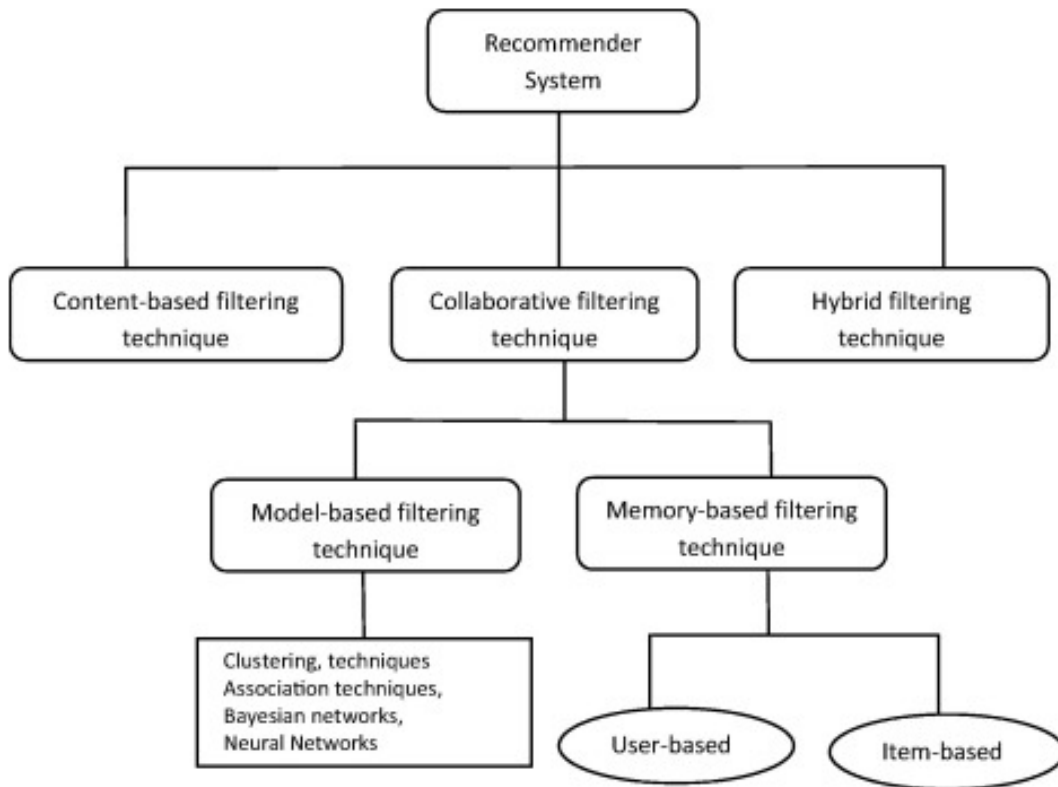


Рисунок 2.1 – Класифікація рекомендаційних алгоритмів

Розглянемо детально існуючі методи задля того щоб виявити який з них найкраще підійде до такої інформаційної системи як туристична установа, з урахуванням тих процесів які в ній відбуваються.

2.1 Методи на основі контенту

Фільтрування на основі контенту використовує функції елемента, щоб рекомендувати інші елементи, подібні до того, що подобається користувачеві, виходячи з попередніх дій або явних відгуків.

Методи на основі контенту - це алгоритми, що залежать від домену, і вони більшою мірою акцентують увагу на аналізі атрибутів елементів для створення

прогнозів. Якщо потрібно рекомендувати такі документи, як веб-сторінки, публікації чи новини, техніка фільтрації на основі контенту є найбільш успішною [6].

У техніці фільтрації на основі контенту рекомендується базуватися на профілях користувачів, використовуючи данні, вилучені із контенту елементів, які користувач оцінював у минулому. Користувачам рекомендуються елементи, які в основному пов'язані з позиціями з позитивною оцінкою. Наприклад, для моделювання взаємозв'язку між різними елементами може використовуватися модель векторного простору або ймовірнісні моделі, такі як Naive Bayes Classifier, дерева рішень або нейронні мережі.

Ці методи дають рекомендації, вивчаючи базову модель за допомогою статистичного аналізу або технічних засобів машинного навчання. Техніка фільтрування на основі контенту не потребує профілю інших користувачів, оскільки вони не впливають на рекомендації. Крім того, якщо змінюється профіль користувача, техніка CBF все ще має потенціал коригувати свої рекомендації за дуже короткий проміжок часу. Основним недоліком цієї методики є необхідність глибокого знання та опису особливостей предметів у профілі.

На відміну від методів колаборативної фільтрації, які покладаються лише на взаємодію між елементами користувача, підходи, засновані на контенті, використовують додаткову інформацію про користувачів та/або елементи. Цією додатковою інформацією можуть бути, наприклад, вік, стать, робота або будь-яка інша особиста інформація для користувачів, а також категорія, основні суб'єкти, тривалість чи інші характеристики [7].

Тоді ідея методів, що базуються на контенті, полягає у спробі побудувати модель на основі наявних функцій, що пояснюють спостережувані взаємодії між елементами користувача.

Методи, що базуються на контенті, страждають від проблеми холодного старту набагато менше, ніж колаборативні підходи: нові користувачі або елементи

можуть бути описані за їх характеристиками, і тому для цих нових об'єктів можна зробити відповідні пропозиції. Від цього недоліку логічне постраждають лише нові користувачі або елементи з раніше небаченими функціями, але як тільки система достатньо стара, на це мало шансів.

Підхід на основі контенту вимагає великої кількості інформації про власні функції предметів, а не використання взаємодії та відгуків користувачів. Схематично цей підхід зображений на рисунку 2.2.

У методах, що базуються на контенті передбачається деяка модель прихованої взаємодії. Вона надається контентом, який визначає представлення користувачів та/або елементів: наприклад, користувачі представлені даними функціями, і ми намагаємось моделювати для кожного елемента той тип профілю користувача, якому подобається цей елемент [8].

Тож, в загальному випадку з застосуванням підходу на базі контенту система порекомендує все, що схоже на предмет, який вам сподобався раніше. На етапі побудови моделі система спочатку знайде схожість між усіма парами елементів а потім використовує найбільш схожі елементи з уже оціненими елементами користувача, щоб сформувати список рекомендацій на етапі рекомендацій.

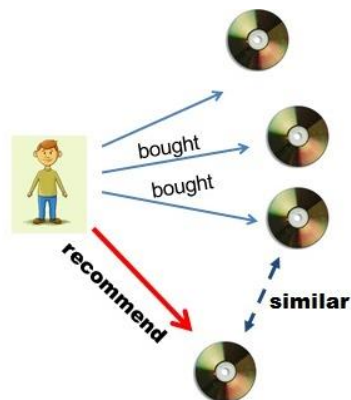


Рисунок 2.2 – Схема рекомендацій на основі контенту

2.3 Методи коллаборативної фільтрації

Методи коллаборативної фільтрації для систем рекомендування - це методи, що ґрунтуються виключно на минулих взаємодіях, зафіксованих між користувачами та предметами з метою створення нових рекомендацій. Ці взаємодії зберігаються у так званій матриці взаємодій між елементами користувача, як зображено на рисунку 2.3.

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

Рисунок 2.3 - Матриця взаємодії

Тоді, основна ідея, яка регулює методи коллаборативної фільтрації, полягає в тому, що ці минулі взаємодії між елементами користувачів є достатніми для виявлення подібних користувачів та/або подібних елементів та для побудови кореляцій між ними. [9].

Клас алгоритмів коллаборативної фільтрації поділяється на дві під категорії, які, як правило, називаються підходами, заснованими на пам'яті та на основі моделі. Підходи, засновані на пам'яті, безпосередньо працюють із значеннями записаних взаємодій, не передбачаючи моделі, а по суті базуються на пошуку найближчих сусідів (наприклад, знайти найближчих користувачів із зацікавленого користувача та запропонувати найпопулярніші елементи серед цих сусідів).

Моделльні підходи передбачають основу "генеративної" моделі, яка пояснює взаємодію між елементами користувача та намагається розкрити її для того, щоб робити нові прогнози. Як і багато методів машинного навчання, система

рекомендацій робить передбачення на основі історичної поведінки користувачів. Зокрема, слід передбачити перевагу користувачів для набору елементів на основі минулого досвіду.

Основна перевага підходів колаборативної фільтрації полягає в тому, що вони не потребують інформації про користувачів або елементи, і, таким чином, вони можуть бути використані у багатьох ситуаціях. Більше того, чим більше користувачів взаємодіє з елементами, тим новіші рекомендації стають точними: для фіксованого набору користувачів та елементів нові взаємодії, записані з часом, приносять нову інформацію та роблять систему більш ефективною.

Однак, оскільки це стосується лише минулих взаємодій для надання рекомендацій, колаборативна фільтрація страждає від "проблеми з холодним запуском": неможливо нічого рекомендувати новим користувачам або рекомендувати новий елемент будь-яким користувачам, а багато користувачів або елементи мають занадто мало взаємодій щоб ефективно поводитися.

Цей недолік можна вирішити по-різному: рекомендувати випадкові елементи новим користувачам або нові елементи випадковим користувачам, рекомендувати популярні елементи новим користувачам або нові елементи найбільш активним користувачам, рекомендувати набір різних елементів для нових користувачів або новий елемент для набору різних користувачів або, нарешті, використання методу, що не працює у співпраці, для нових користувачів або елементів [10].

Стандартний метод колаборативної фільтрації відомий як алгоритм найближчого сусідства. Моделі ґрунтуються на припущенні, що людям подобаються речі, подібні до інших, які їм подобаються, і речі, які подобаються іншим людям, як показано на рисунку 2.4.

В основному ідея полягає в тому, щоб знайти найбільш схожих користувачів з вашим цільовим користувачем (найближчими сусідами) і оцінити їх рейтинг для предмета як прогнозування рейтингу цього елемента для цільового користувача.

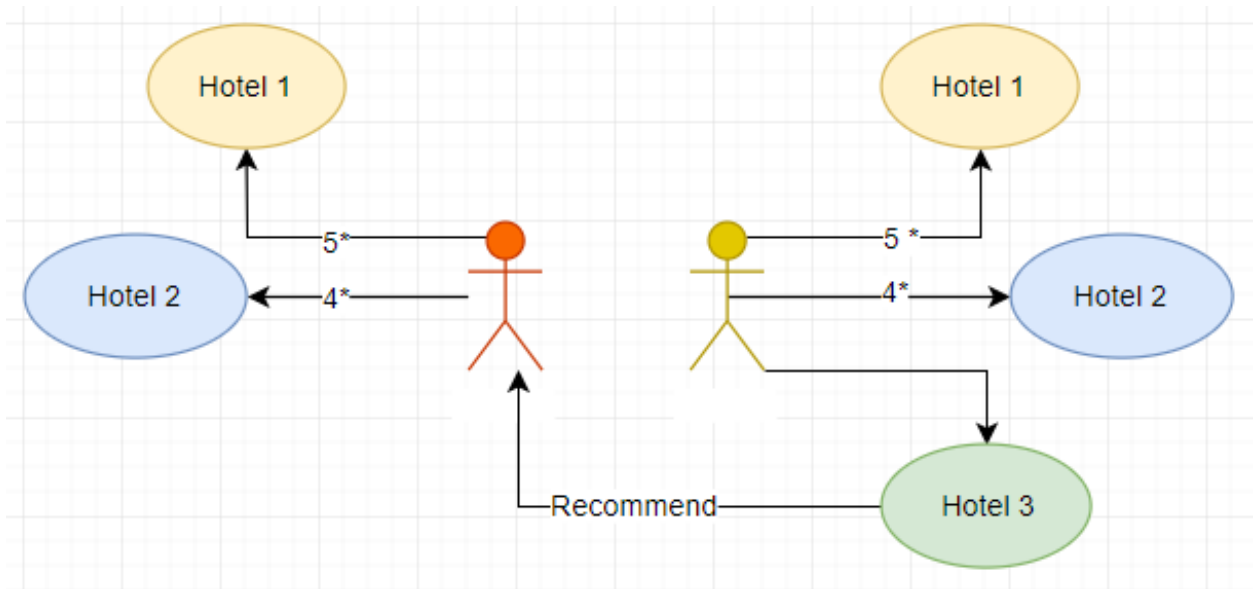


Рис. 2.4 - Схема рекомендацій за колаборативної фільтрації

В основному, існує два типи такої фільтрації. Це модельний підхід та підхід на основі пам'яті.

Підходи колаборативного фільтрування на основі пам'яті можна розділити на два основні розділи: фільтрування предметів користувача та фільтрація елементів. Фільтрація елементів користувача бере конкретного користувача, знаходить користувачів, подібних до цього користувача на основі схожості оцінок, і рекомендує елементи, які сподобалися цим схожим користувачам. На відміну від цього, фільтрування елементів-предметів знайде предмет, знайде користувачів, які сподобалися, і знайде інші предмети, які сподобалися цим користувачам або подібним користувачам. Він приймає елементи та виводить інші елементи як рекомендації.[11]

Ключова відмінність підходу, заснованого на пам'яті, від методики, заснованої на моделі полягає в тому, що ми не вивчаємо жоден параметр за допомогою градієнтного спуску (або будь-якого іншого алгоритму оптимізації). Найближчий користувач або елементи обчислюються лише за допомогою

косинусної подібності або коефіцієнтів кореляції Пірсона, які базуються лише на арифметичних операціях.

Коллаборативна фільтрація не потребує нічого іншого, крім історичних уподобань користувачів щодо набору елементів. Оскільки вона заснована на історичних даних, основне припущення тут полягає в тому, що користувачі, які домовилися в минулому, як правило, також погоджуються в майбутньому.

З точки зору переваг користувача, це зазвичай виражається двома категоріями. Явна оцінка - це показник, який користувач надає предмету ковзаючої шкали, як 5 зірок. Це найбільш прямий відгук користувачів, щоб показати, наскільки їм сподобався товар.

У методах коллаборативної фільтрації на основі пам'яті не передбачається прихованої моделі. Алгоритми безпосередньо працюють із взаємодіями між елементами користувача: наприклад, користувачі представлені їх взаємодією з елементами, а пошук найближчих сусідів у цих представленнях використовується для отримання пропозицій. Оскільки не передбачається прихованої моделі, ці методи теоретично мають низький ухил, але високу дисперсію.

У методах коллаборативної фільтрації, заснованих на моделі, передбачається деяка модель прихованої взаємодії. Модель навчається реконструювати значення взаємодій між елементами користувачів із власного представлення користувачів та елементів. Потім на основі цієї моделі можна зробити нові пропозиції. Користувачі та елементи, що виховуються за допомогою моделі, мають приховані уявлення про математичне значення, яке може бути важко інтерпретувати для людини. Оскільки передбачається (досить вільна) модель взаємодії між користувачами, цей метод теоретично має більше зміщення, але меншу дисперсію, ніж методи, що передбачають відсутність прихованої моделі.[12]

Оскільки тренування чи оптимізація не передбачається, цей підхід простий у використанні. Але його ефективність знижується, коли ми маємо розрізнені дані,

що перешкоджають масштабованості такого підходу для більшості проблем у реальному світі.

Коллаборативна фільтрація забезпечує сильну прогностичну потужність для систем рекомендування та вимагає при цьому найменшої інформації.

Однак у деяких конкретних ситуаціях є обмеження. По-перше, основні смаки, виражені прихованими ознаками, насправді не інтерпретуються, оскільки немає властивостей метаданих, пов'язаних із змістом. З іншого боку, коллаборативна фільтрація стикається з холодним початком. Коли новий елемент надходить, поки його не оцінить значна кількість користувачів, модель не може дати персоналізованих рекомендацій. Так само для предметів із хвоста, які не отримали занадто багато даних, модель, як правило, надає їм меншої ваги та має зміщення популярності, рекомендуючи більш популярні предмети.

Коллаборативна фільтрація має ряд недоліків і один з них це проблема холодного старту. Це стосується ситуації, коли рекомендуючий не має належної інформації про користувача чи предмет для того, щоб зробити відповідні прогнози. Це одна з найважливіших проблем, яка знижує ефективність системи рекомендацій. Профіль такого нового користувача або елемента буде порожнім, оскільки він не оцінив жоден елемент; отже, його смак не відомий системі.

Окрім цього існує проблема зменшення даних. Це проблема, яка виникає внаслідок нестачі достатньої кількості інформації, тобто коли користувачі оцінюють лише деякі із загальної кількості наявних у базі даних елементів. Це завжди призводить до розрідженої матриці елементів користувача, неможливості знайти успішних сусідів і, нарешті, до створення слабких рекомендацій. Крім того, розрідженість даних завжди призводить до проблем із охопленням, який становить відсоток елементів у системі, щодо яких можна давати рекомендації.

Не варто забувати і про складність масштабованості. Це ще одна проблема, пов'язана з алгоритмами рекомендацій, оскільки обчислення зазвичай зростають лінійно з кількістю користувачів та елементів. Техніка рекомендацій, яка є

ефективною, коли кількість наборів даних обмежена, можливо, не буде ефективною при збільшенні обсягу набору даних. Таким чином, важливо застосовувати методи рекомендацій, які здатні успішно масштабувати, оскільки кількість наборів даних у базі даних збільшується. Методи, що застосовуються для вирішення проблеми масштабування та прискорення формування рекомендацій, базуються на методах зменшення розмірності, таких як метод синхронного розкладання величини (SVD), який має можливість виробляти надійні та ефективні рекомендації.[13]

І на останок колаборативна фільтрація може мати схильність до синонімії. Синонімія - це тенденція дуже схожих предметів мати різні назви чи записи. Більшість систем рекомендацій важко розрізнити тісно пов'язані між собою елементи, наприклад, різницю між, дитячим одягом та дитячою тканиною. Системи спільної фільтрації зазвичай не знаходять відповідності між двома термінами, щоб можна було обчислити їх схожість. Різні методи, такі як автоматичне розширення термінів, побудова тезауруса і сингулярне розкладання значення (SVD), особливо латентна семантична індексація, здатні вирішити проблему синонімії. Недолік цих методів полягає в тому, що деякі додані терміни можуть мати різні значення від призначеного, що іноді призводить до швидкої деградації виконання рекомендацій.

2.3 Гібридні методи

Гібридні системи рекомендацій налаштовані для з'єднання підходів на основі контенту та колаборативної фільтрації які можливо застосувати на одній предметній області, і збільшують переваги, а також зменшують слабкі сторони обох методів.

Саме тому гібридні рекомендаційні системи працюють на характеристиках, які пов'язані з обома підходами. Дійсно, є багато підходів, які ми можемо об'єднати на основі контенту з колаборативною фільтрацією (CF) [14].

Як фільтрування на основі контенту, так і колаборативне фільтрування мають сильні та слабкі сторони. Для фільтрування на основі контенту можна виділити три конкретні проблеми. По перше створити корисний опис контенту у деяких областях може бути досить складно. Наприклад, у доменах, де елементи складаються з музики чи відео, подання контенту не завжди можливо за допомогою сучасної технології.

По друге це складна спеціалізація. Система фільтрації на основі контенту не буде вибирати елементи, якщо попередня поведінка користувача не підтверджує це. Необхідно додати додаткові методи, щоб надати системі можливість висловлювати пропозиції поза межами того, до чого користувач уже виявив інтерес.

Також ситуацію ускладнює суб'єктивна проблема домену. Методи фільтрації на основі контенту мають труднощі розрізнити суб'єктивну інформацію, таку як точки зору та гумор.

Коллаборативна система фільтрації не має цих недоліків. Оскільки немає необхідності в описі елементів, що рекомендуються, система може обробляти будь-яку інформацію. Крім того, система здатна рекомендувати користувачеві елементи, які можуть мати зовсім різний зміст, ніж ті, що зацікавили користувача раніше. Нарешті, оскільки рекомендації ґрунтуються на думці інших, вони добре підходять для суб'єктивних областей, наприклад, таких як мистецтво. Однак колаборативна фільтрація також має певні проблеми.

Одна з них це рання проблема з рейтингами. Системи колаборативної фільтрації не можуть надати рекомендації щодо нових елементів, оскільки немає рейтингів користувачів, на яких можна базувати прогноз. Навіть якщо користувачі почнуть оцінювати товар, пройде якийсь час, перш ніж товар отримає достатню кількість оцінок, щоб дати точні рекомендації. Аналогічно, рекомендації також будуть неточними для нових користувачів, які оцінили мало елементів.

По-друге, існує проблема зрідженості. У багатьох інформаційних сферах наявна кількість предметів перевищує кількість, яку людина здатна (і бажає)

дослідити на сьогоднішній день. Це ускладнює пошук предметів, оцінених достатньою кількістю людей, на яких можна базувати прогнози.

І на останок, щоб усе працювало коректно повинні бути групи користувачів з характеристиками що перетинаються. Навіть якщо такі групи існують, люди, які не погоджуються або не погоджуються ні з одною групою людей, отримуватимуть неточні рекомендації.

Система, що поєднує фільтрацію на основі контенту та колаборативну фільтрацію, може скористатися як представленням контенту, так і подібністю серед користувачів. Хоча існує декілька способів поєднання двох методів, можна розрізнити два базові підходи. Гібридний підхід поєднує в собі два типи інформації, хоча також можна використовувати рекомендації двох методів фільтрації незалежно.

Підсумовуючи можна зробити висновок що колаборативна фільтрація шукає співвідношення між рейтингами користувачів для прогнозування. Таке співвідношення є найбільш значимим, коли користувачі мають багато спільних позицій. Як було сказано раніше, у великих областях з багатьма предметами це не завжди так. Крім того, відсутність доступу до контенту елементів перешкоджає збігу подібних користувачів, якщо вони не оцінили такий самий елемент. Наприклад, якщо одному користувачеві сподобався фільм «Роккі», а іншому сподобався фільм «Рокі II», вони не обов'язково будуть узгоджуватися разом. Гібридний підхід, який називається співпрацею за допомогою контенту, займається цими питаннями, включаючи як інформацію, що використовується під час фільтрування на основі контенту, так і колаборативної фільтрації.[15]

У співпраці за допомогою контенту для створення профілю користувача використовуються як рейтингові елементи, так і зміст елементів. Вибір термінів, які описують зміст предметів, здійснюється за допомогою контентних методик. Вага термінів вказує на те, наскільки вони важливі для користувача.

Як і при колаборативній фільтрації, для обчислення кореляції між користувачами можна використовувати коефіцієнт кореляції Пірсона. Замість визначення кореляції за рейтингами користувачів, можуть використовуватися терміни ваги. Оскільки у цього методу є більша кількість елементів, за допомогою яких можна визначити схожість, ніж з застосуванням колаборативної фільтрації, проблема користувачів, які не мають достатньо загальних позицій, вже не є проблемою. Крім того, на відміну від фільтрування на основі контенту, прогнози ґрунтуються на враженнях інших користувачів, які можуть призвести до рекомендацій поза звичайним середовищем користувача.

Однак для надання рекомендацій щодо товарів все-таки необхідно, щоб було достатньо користувачів, які оцінили товар. Так само, як і в колаборативній фільтрації, нові елементи не можна рекомендувати, якщо немає користувачів, які оцінили цей елемент.

Інший підхід до поєднання колаборативної та контентної фільтрації полягає у складанні прогнозів на основі середньозваженої середньої рекомендації на основі контенту та рекомендації щодо спільної роботи. Ранг кожного рекомендованого предмета може бути мірою для ваги. Таким чином найвища рекомендація отримує найбільшу вагу.

За підсумками детального аналізу існуючих методів побудови рекомендаційних системи та виявленими перевагами з недоліками кожного з підходів можна зробити наступний висновок [16].

В контексті заданої предметної області, а саме туристичної установи, онлайн сервіс якої дозволяє бронювати готелі здається доречним для побудови рекомендаційного механізму використовувати саме колаборативну фільтрацію.

Основоположним чинником є специфіка предметної області, бо готелі самі по собі можуть мати схожі характеристики але в житті являти собою зовсім різні установи. Саме тому доречним буде будувати рекомендаційний алгоритм базуючись на кореляції користувачів аніж аналізу контенту.

2.4 Дослідження механізму колаборативної фільтрації

Коллаборативна фільтрація - це технологія прогнозування, що не залежить від домену, та не може бути легко і адекватно описана метаданими. Техніка колаборативної фільтрації працює за допомогою побудови бази даних (матриця-елемент-користувач) налаштувань елементів для користувачів. Потім вона відповідає користувачам за відповідними інтересами та уподобаннями, обчислюючи схожість між їх профілями для надання рекомендацій, тобто побудови кореляції. Такі користувачі будують групу під назвою сусідство (рисунок 2.5). Користувач отримує рекомендації щодо тих предметів, які він раніше не оцінював, але які вже були позитивно оцінені користувачами подібними до нього (що знаходяться поряд в матриці). [17]

Рекомендації не варто плутати з передбаченнями. Передбачення - це числове значення, R_{ij} , що виражає передбачуваний бал пункту j для користувача i , тоді як рекомендація - це список найпопулярніших N елементів, які користувачеві найімовірніше найбільше сподобаються. Техніка колаборативної фільтрації може бути поділена на дві категорії: на основі пам'яті та на основі моделей.

Алгоритми на основі пам'яті підходять до проблеми спільної фільтрації за допомогою всієї бази даних. Вони намагаються знайти користувачів, схожих на активного користувача (тобто користувачів, для яких ми хочемо робити прогнози), і використовують їх уподобання для прогнозування рейтингів для активного користувача.

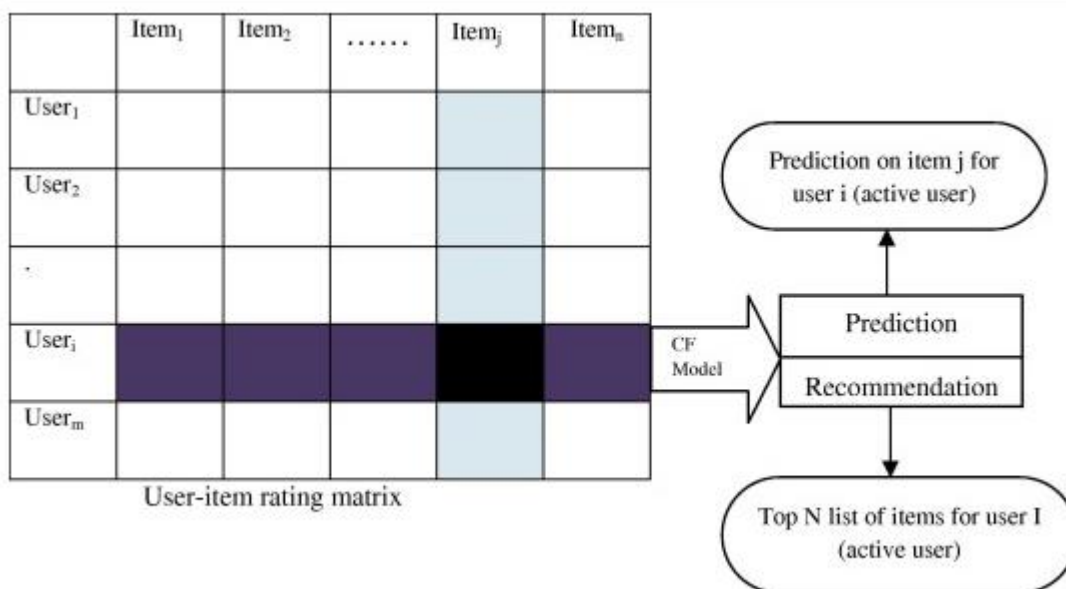


Рисунок 2.5 - Процес колаборативної фільтрації

Перш ніж потрапити до матриці елементи оцінюються користувачами. Після того, як буде знайдений сусід користувача, різні алгоритми можуть використовуватися для об'єднання уподобань сусідів задля створення рекомендацій. Завдяки ефективності цих методів вони досягли широкого успіху застосування в реальному житті. CF на основі пам'яті може бути досягнута двома способами на основі користувацьких та предметних методів. Метод колаборативної фільтрації, на базі кореляції користувачів, обчислює схожість між користувачами, порівнюючи їх рейтинги на тому самому елементі, а потім обчислює передбачуваний рейтинг для активного користувача як середньозважене середнє значення рейтингів товару, подібне до активного користувача, де ваги - це подібність цих користувачів з цільовим елементом. Методи фільтрування на основі предметів обчислюють прогнози, використовуючи подібність між елементами, а не подібність між користувачами. Вони будують модель подібності елементів шляхом видалення всіх елементів, оцінених активним користувачем з матриці елементів користувача, та визначають наскільки схожі отримані елементи цільовому елементу. Потім

вибирається k найбільш подібних. Прогнозування здійснюється шляхом взяття середньозваженого показника активних користувачів на аналогічні елементи k . Для обчислення подібності між елементом/користувачем використовується кілька типів подібності. Два найпопулярніші заходи подібності - це кореляція та косинус.

Кореляція - це двовимірний аналіз, який вимірює силу асоціації між двома змінними та напрям зв'язку. З точки зору сили взаємозв'язку значення коефіцієнта кореляції коливається між $+1$ та -1 . Значення ± 1 вказує на ідеальну ступінь асоціації між двома змінними. Оскільки значення коефіцієнта кореляції йде в бік 0 , відношення між двома змінними буде слабкішим. Напрямок зв'язку позначається знаком коефіцієнта; знак $+$ позначає позитивне відношення, а знак $-$ позначає негативне відношення [18].

Для вимірювання подібності ми хочемо знайти співвідношення між двома користувачами. Це дає нам певне значення, яке визначає, наскільки схожі два користувачі. Максимальне означає, що вони обидва оцінюють абсолютно однаково, тоді як мінімальне значення означає, що вони оцінюють речі прямо протилежно.

Одним з найвідоміших засобів є коефіцієнт кореляції Пірсона (рисунок 2.6). Це основний алгоритм кореляції для зразків, адаптованих до рейтингової інформації. Він намагається виміряти, наскільки двоє користувачів різняться разом від їх нормальних передбачень - тобто напрям/величина кожного вибору порівняно із середнім рівнем їх виборів. Якщо вони змінюються однаково за предметами, які вони оцінили спільно, вони отримають позитивну кореляцію; інакше вони отримають негативну кореляцію.

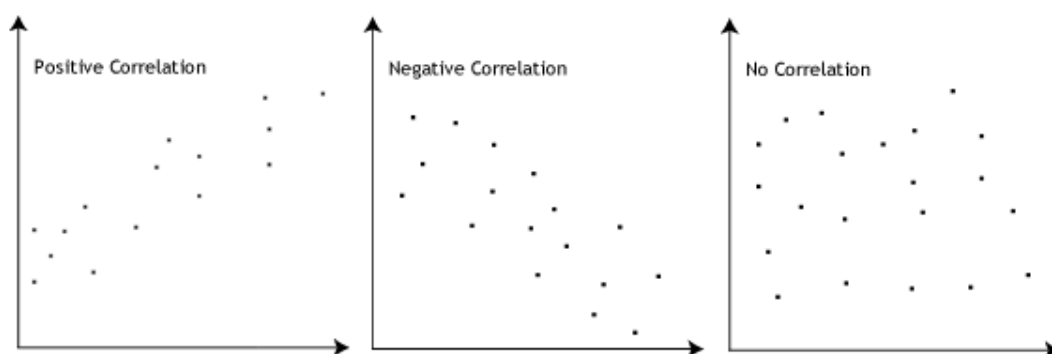


Рисунок 2.6 - Кореляція за Пірсоном

Коефіцієнт кореляції Пірсона - це тестова статистика, яка вимірює статистичну залежність або асоціацію між двома безперервними змінними. Він відомий як найкращий метод вимірювання асоціації між залежними змінними, оскільки він заснований на методі коваріації. Він дає інформацію про величину асоціації, або кореляцію, а також напрямок відносин.

Знаходячи кореляцію треба завжди пам'ятати що по-перше, кореляція ніколи не буває нижче -1 . Кореляція рівна -1 вказує на те, що точки даних на графіку розкидання лежать саме на прямій низхідній лінії; дві змінні повністю негативно лінійно пов'язані. Кореляція 0 означає, що дві змінні взагалі не мають лінійного відношення, як показано на рисунку 2.7. Однак може існувати деяке нелінійне відношення між двома змінними. Також варто пам'ятати що коефіцієнти кореляції ніколи не перевищують 1 . Коефіцієнт кореляції рівний 1 означає, що дві змінні є ідеально позитивно лінійно пов'язаними; точки на розсіяному графіку лежать рівно на прямій висхідній лінії.

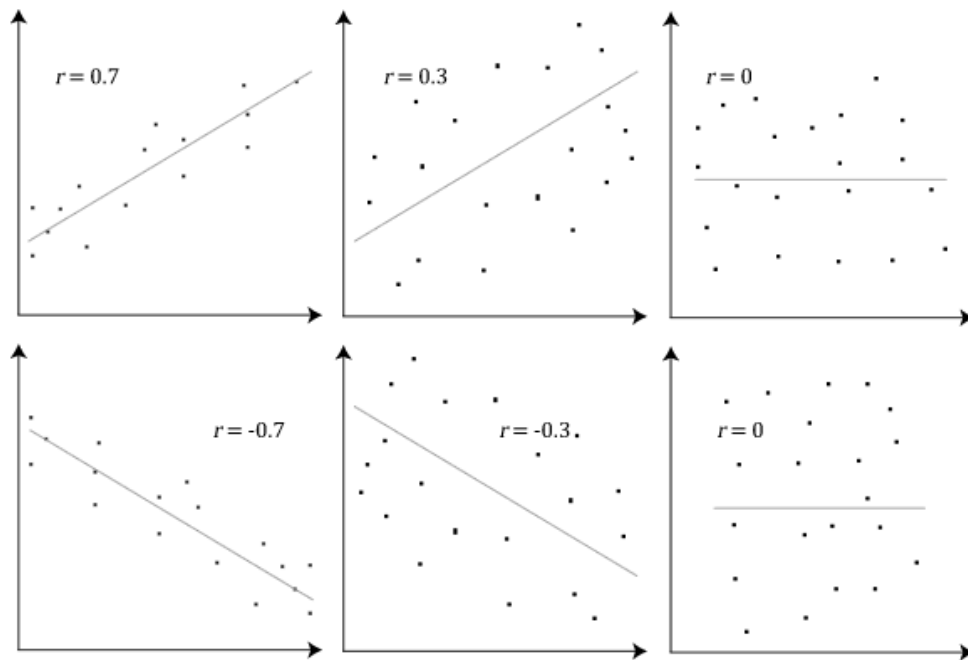


Рисунок 2.7 - Приклади кореляції

Інше вимірювання подібності називається векторною подібністю. У векторній формі ми можемо представити кожну змінну у вигляді серії координат, причому кожна з них вказує на позицію в іншому просторовому вимірі. Якщо вектор має два елементи, тобто він має лише дві змінні, робота з ним подібна до перевірки позиції елемента на карті, використовуючи перше число для позиції на осі X , а друге на осі Y (рисунок 2.8 [19]).

Кожен приклад на рисунку (крапка на графіку) це упорядкований список значень (званий кортеж), який легко розміщується та друкується на просторі, використовуючи перше значення списку для X та друге для Y .

Якщо набір даних у матричній формі має безліч числових ознак (стовпців), в ідеалі кількість ознак являє собою розміри простору даних, тоді як рядки представляють кожну точку, яка математично є вектором. Коли у вектора більше двох елементів, візуалізація стає клопіткою, тому що представляти розмірність вище третього непросто.

Однак ми можемо додати більше розмірності за допомогою деяких доцільних, наприклад, використовуючи розмір, форму або колір для інших характеристик. Зрозуміло, що це непросте завдання, і часто результат далеко не інтуїтивний. Однак ми можемо зрозуміти, де би знаходилися точки у просторі даних, систематично друкуючи багато графіків, розглядаючи розміри два на два. Такі графіки називаються матрицями розсіпу.

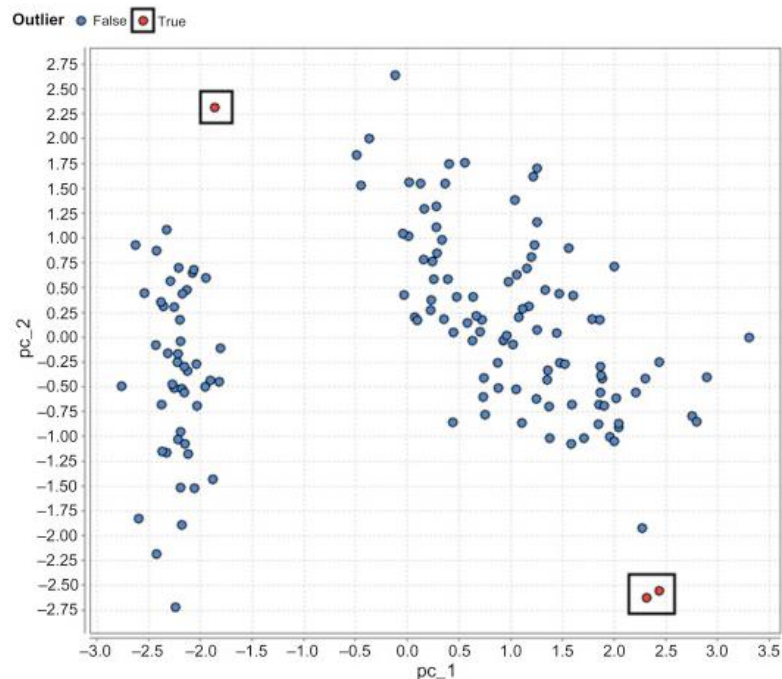


Рисунок 2.8 - Векторна подібність

Ми можемо розглядати двох користувачів як вектори у n -мірному просторі, де n - кількість елементів у базі даних. Як і будь-які два вектори, ми можемо порівняти кут між ними. Інтуїтивно, якщо два вектори, як правило, вказують в одному напрямку, вони отримують позитивну схожість; якщо вони вказують у протилежні сторони, вони отримують негативну схожість. Щоб імітувати це, ми просто беремо кут косинуса між цими двома векторами, що дає нам значення від -1 до 1.

Для колаборативної фільтрації на основі користувачів, подібність двох користувачів вимірюється як косинус кута між векторами. Для користувачів u та u' схожість косинуса:

$$sim(u, u') = \cos(\theta) = \frac{r_u * r_{u'}}{\|r_u\| * \|r_{u'}\|}$$

Ми можемо передбачити рейтинг користувача для готелю, взявши зважену суму оцінок готелів у всіх інших користувачів (у яких) зважування - це кількість подібності між кожним користувачем та користувачем, та нормалізувати її:

$$\hat{r}_{ui} = \sum_{u'} sim(u, u') * r_{u'i}$$

Оскільки тренування чи оптимізація не передбачається, це підхід простий у використанні. Але його ефективність знижується, коли ми маємо розрізнені дані, що перешкоджають масштабованості такого підходу для більшості проблем у реальному світі.

Найважливішим компонентом систем рекомендацій, є визначення схожості користувачів та вибір найближчих сусідів. Процес складеться з трьох фаз. У фазі 1, номери позицій, ідентифікуються за допомогою визначення елементів, оцінених активним користувачем та іншими користувачами. Це робиться шляхом сортування елементів. Наступний крок - ідентифікація. У великій і розрідженій матриці важко ідентифікувати окремі елементи, у цьому разі має сенс розгрупувати їх.

Фаза 2 - це пошук подібних користувачів. Перший крок – це вилучити несхожих користувачів. Виключення несхожих користувачів усуне галасливих користувачів які не мають значення для цього процесу і полегшить проблему зрідженості. Наступний крок – класифікація подібних користувачів. Після цього

необхідно обчислити схожість користувачів за допомогою міри подібності. Останній крок - визначення значення подібності.

Фаза 3 - це знаходження найближчих сусідів. До них відносять найбільш схожих користувачів, які впливають на фактор прогнозування рейтингу. У цій фазі отримане значення подібності сортується на основі найвищого значення. Користувачі з високим значенням подібності вважаються найбільш схожими користувачами. Це користувачі з високим рівнем позитивної кореляції.

Наступним кроком є визначення порогу подібності для визначення кількості K найближчих сусідів. Результат цього процесу вдосконалення конкретно значення подібності, яке буде використано для отримання рейтингу прогнозу та генерації N рекомендацій. На рисунку 2.9 зображено цей процес.

В загальному виді алгоритм має наступні кроки:

- побудова матриці елементів користувача;
- визначення та групування об'єктів, що оцінюються разом;
- класифікація подібних користувачів та видалення обмежень;
- обчислення подібності користувачів;
- сортування значення подібності;
- вибір найбільш схожих користувачів;
- встановлення порогу подібності;
- знаходження найближчих сусідів;
- обчислення рейтингу прогнозу та отримання рекомендацій.

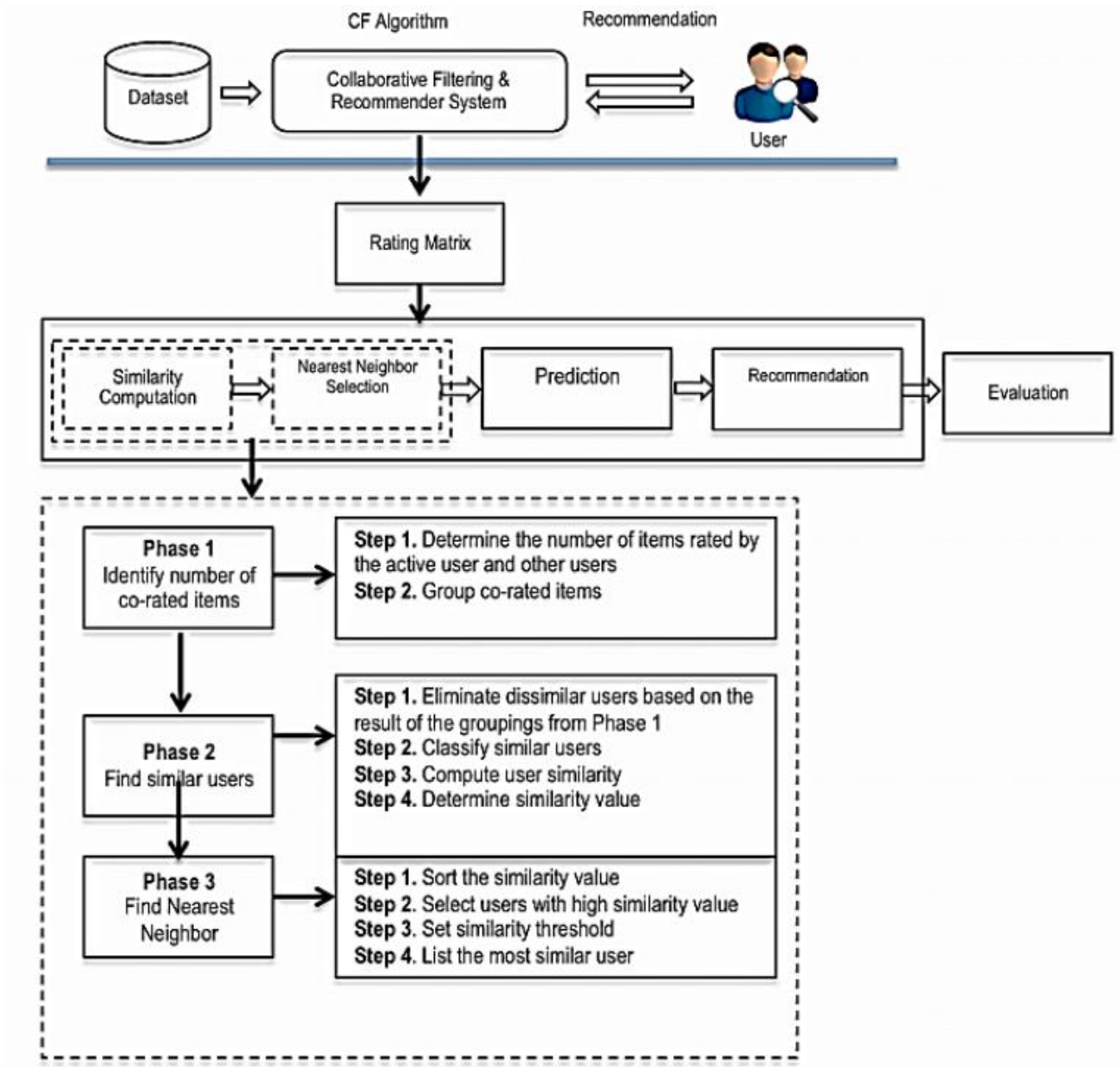


Рисунок 2.9 - Процес колаборативної фільтрації.

Перший крок – формування матриці елементів користувача. Матриця зображена на рисунку 2.10. У ній показано рейтинг користувачів на елементах від 1-5. Нульове значення означає, що користувач не оцінював предмет і вважається рідкісними даними.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
User 1	2	3	4	4	5	2	3	3	5	3
User 2	3	3	5	4	3	4	5	2	4	4
User 3	5	5	1	5	1	5	3	4	1	0
User 4	0	2	0	1	2	1	1	0	2	4
User 5	1	1	2	0	3	0	5	1	4	1
User 6	4	4	3	3	0	1	5	3	3	2
User 7	0	0	5	4	3	3	1	5	4	3
User 8	2	2	3	5	4	5	3	2	1	4
User 9	3	3	1	3	5	4	2	5	2	0
User 10	5	4	2	0	2	5	2	0	3	3
Active User	0	2	4	3	0	2	4	1	0	0

Рисунок 2.10 – Матриця елемент користувач

Далі визначається кількість предметів з номінальною оцінкою. Елементи, оцінені активним користувачем, визначаються шляхом сортування. Результат зображено на рисунку 2.11.

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	Item 1	Item 5	Item 9	Item 10
User 1	4	3	4	3	2	3	2	5	5	3
User 2	5	5	4	3	4	2	3	3	4	4
User 3	1	3	5	5	5	4	5	1	1	0
User 4	0	1	1	2	1	0	0	2	2	4
User 5	2	5	0	1	0	1	1	3	4	1
User 6	3	5	3	4	1	3	4	0	3	2
User 7	5	1	4	0	3	5	0	3	4	3
User 8	3	3	5	2	5	2	2	4	1	4
User 9	1	2	3	3	4	5	3	5	2	0
User 10	2	2	0	4	5	0	5	2	3	3
Active User	4	4	3	2	2	1	0	0	0	0

Рисунок 2.11 – Відсортована матриця елемент користувач

Щоб знайти подібних користувачів, видаляються різні користувачі та групи предметів без оцінки, щоб зняти шум. Наступним кроком є класифікація подібних користувачів. На рисунку 2.12 зображена матриця без зайвих елементів.

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	Item 1	Item 5	Item 9	Item 10
User 1	4	3	4	3	2	3	2	5	5	3
User 2	5	5	4	3	4	2	3	3	4	4
User 3	1	3	5	5	5	4	5	1	1	0
User 6	3	5	3	4	1	3	4	0	3	2
User 8	3	3	5	2	5	2	2	4	1	4
User 9	1	2	3	3	4	5	3	5	2	0
Active User	4	4	3	2	2	1	0	0	0	0

Рисунок 2.12 –Матриця елемент користувач після видалення шуму

Наступний крок - визначення найбільш подібних користувачів за допомогою схожості косинуса. Результат зображений на рисунку 2.13.

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	PCC
User 1	4	3	4	3	2	3	0.51189
User 2	5	5	4	3	4	2	0.94176
User 3	1	3	5	5	5	4	-0.6528
User 6	3	5	3	4	1	3	0.41416
User 8	3	3	5	2	5	2	0.20146
User 9	1	2	3	3	4	5	-0.9342
Active User	4	4	3	2	2	1	

Рисунок 2.13 –Матриця з підрахованою мірою кореляції

У пошуку найближчих сусідів користувачі з високим значенням подібності вважаються найбільш подібними. Це користувачі з високою позитивною кореляцією. На цьому кроці обираються K найближчих сусідів. Результат наведений на рисунку 2.14.

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8
User 1	4	3	4	3	2	3
User 2	5	5	4	3	4	2
User 6	3	5	3	4	1	3
User 8	3	3	5	2	5	2

Рисунок 2.14 - Матриця найближчих сусідів

Для обчислення рейтингу прогнозу, до середньозваженого значення серед активних користувачів додається середнє відхилення від сусідів. Відхилення використовуються для коригування упереджених значень. Упередження виникають, коли деякі користувачі схильні давати високі або низькі оцінки всім предметам. Стандартна метрика прогнозування має вигляд:

$$P_{a,i} = r'_a + \frac{\sum_{u \in K} (r_{u,i} - r'_u) \cdot W_{a,u}}{\sum_{u \in K} W_{a,u}}$$

Отримана матриця зображена на рисунку 2.15.

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	Item 1	Item 5	Item 9	Item 10
User 1	4	3	4	3	2	3	2	5	5	3
User 2	5	5	4	3	4	2	3	3	4	4
User 6	3	5	3	4	1	3	4	0	3	2
User 8	3	3	5	2	5	2	2	4	1	4
Active User	4	4	3	2	2	1				

Рисунок 2.15 - Матриця прогнозу

Виходячи з результату, прогнозований рейтинг для пункту 9 має значення найвище значення - 3,37, а за ним пункти 10 - 3,23, пункт 5 з 3.07, а пункт 1 - з 2.73.

Загалом виділяють два методи колаборативної фільтрації – користувач-користувач та елемент-елемент. Для того, щоб зробити нову рекомендацію для користувача, метод користувач-користувач грубо намагається визначити користувачів з найбільш схожим "профілем взаємодії" (найближчими сусідами), щоб запропонувати елементи, які є найпопулярнішими серед цих сусідів. Цей метод "орієнтований на користувачів", оскільки він представляє їх на основі взаємодії з елементами та оцінки відстаней між користувачами.

Припустимо, що ми хочемо дати рекомендацію для даного користувача. По-перше, кожен користувач може бути представлений своїм вектором взаємодій з різними елементами. Тоді ми можемо обчислити певну «схожість» між нашим користувачем, що цікавить, та всіма іншими користувачами. Цей показник подібності такий, що двох користувачів із однаковою взаємодією на одних і тих же елементах слід вважати близькими. Після того, як схожість з усіма користувачами була обчислена, ми можемо зберегти k-найближчих сусідів до нашого користувача, а потім запропонувати найпопулярніші елементи серед них.

Щоб зробити нову рекомендацію користувачеві, ідея методу-елемента полягає у пошуку елементів, подібних до тих, з якими користувач уже «позитивно». Два елементи вважаються схожими, якщо більшість користувачів, які взаємодіяли з ними обома, робили це аналогічно. Кажуть, що цей метод "орієнтований на предмет", оскільки він представляє елементи, засновані на взаємодії, які користувачі мали з ними, і оцінює відстані між цими елементами.

Припустимо, що ми хочемо дати рекомендацію для даного користувача. По-перше, ми вважаємо предмет, який цьому користувачу найбільше сподобався, і представляємо його (як і всі інші елементи) вектором взаємодії. Тоді ми можемо обчислити схожість між «найкращим предметом» та всіма іншими предметами. Після того, як схожість була обчислена, ми можемо зберегти k-найближчих сусідів

до вибраного «найкращого продукту», який є новим для нашого користувача, що цікавить, і рекомендуємо ці елементи.

Для отримання більш релевантних рекомендацій, ми можемо виконати цю роботу не тільки для улюбленого елемента користувача та замість цього розглядати п бажаних елементів. У цьому випадку ми можемо рекомендувати предмети, близькі до кількох із цих бажаних предметів.

Метод користувач-користувач заснований на пошуку подібних користувачів з точки зору взаємодії з елементами. Оскільки, як правило, кожен користувач взаємодіяв лише з кількома елементами, це робить метод досить чутливим до будь-яких записаних взаємодій (велика дисперсія). З іншого боку, оскільки остаточна рекомендація базується лише на взаємодії, записаній для користувачів, схожих на нашого користувача, що цікавить, ми отримуємо більш персоналізовані результати.

І навпаки, метод елемента заснований на пошуку подібних елементів з точки зору взаємодії між елементами користувача. Оскільки, як правило, багато користувачів взаємодіяли з елементом, пошук в районі набагато менш чутливий до одинарних взаємодій (нижча дисперсія). В якості контрагента, взаємодія, яка надходить від усіх типів користувачів, потім розглядається в рекомендації, що робить метод менш персоналізованим (більш упередженим). Таким чином, цей підхід менш персоналізований, ніж підхід користувач-користувач, але більш надійний.

Один з найбільших недоліків колаборативної фільтрації на основі пам'яті полягає в тому, що вони не масштабуються легко: створення нових рекомендацій може бути дуже трудомістким для великих систем. Дійсно, для систем з мільйонами користувачів і мільйонами предметів крок пошуку найближчих сусідів може стати незрозумілим, якщо не буде ретельно розробленим. Для того, щоб зробити обчислення більш простежуваними для величезних систем, ми можемо скористатися розрізненістю матриці взаємодії під час проектування нашого алгоритму або використовувати методи приблизних найближчих сусідів.

В свою чергу підходи для колаборативної фільтрації на основі моделей покладаються лише на інформацію про взаємодію користувачів та передбачають приховану модель, яка повинна пояснювати ці взаємодії. Наприклад, алгоритми матричної факторизації полягають у декомпозиції величезної та розрідженої матриці взаємодії між елементами користувача на добуток двох менших і щільних матриць: матриця фактора користувача (містить представлення користувачів), множиться на матрицю елементів-фактора (містить представлення елементів) . Методи на основі матричної факторизації намагаються зменшити розмірність матриці взаємодії та наблизити її за допомогою двох або більше малих матриць із k латентними компонентами, як зображено на рисунку 2.16.

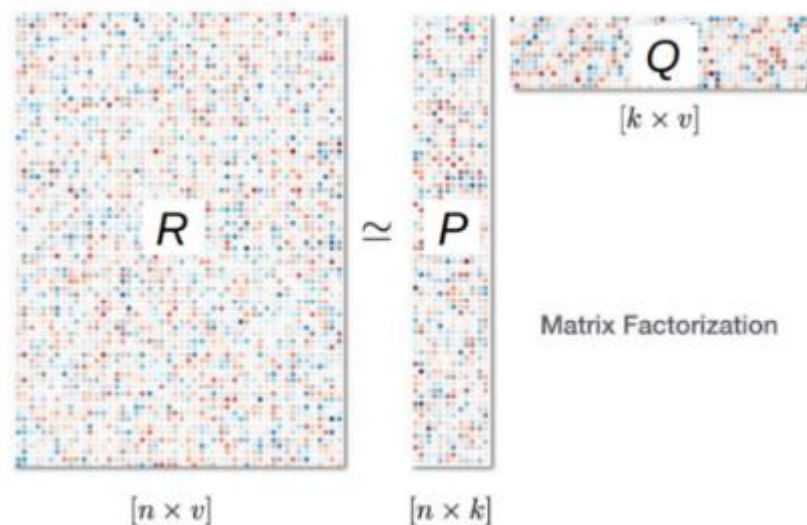


Рисунок 2.16 - Матрична факторизація

В загальному випадку ця методика використовує попередні рейтинги, щоб вивчити модель, щоб поліпшити продуктивність технології колаборативної фільтрації. Процес побудови моделі може здійснюватися за допомогою машинного навчання або методів обміну даними. Ці методи можуть швидко порекомендувати набір елементів за те, що вони використовують попередньо обчислені моделі, і вони

довели, що вони дають результати рекомендацій, схожі на методики рекомендацій на основі сусідства.

Приклади цих методів включають методику зменшення розмірності, такі як сингулярне розкладання величини (SVD), техніка завершення матриці, методи прихованої семантики, регресія та кластеризація. Методи на основі моделей аналізують матрицю елементів користувача для виявлення взаємозв'язків між елементами; вони використовують ці відносини для порівняння списку рекомендацій. Методи, засновані на моделях, вирішують проблеми з обмеженими можливостями, пов'язані з рекомендаційними системами.

2.5 Висновки за розділом 2

За підсумками детального аналізу існуючих методів побудови рекомендаційних системи можна зробити наступний висновок. В контексті заданої предметної області, а саме туристичної установи, онлайн сервіс якої дозволяє бронювати готелі здається доречним для побудови рекомендаційного механізму використовувати саме колаборативну фільтрацію. Основоположним чинником є специфіка предметної області, бо готелі самі по собі можуть мати схожі характеристики але в житті являти собою зовсім різні установи. Саме тому доречним буде будувати рекомендаційний алгоритм базуючись на кореляції користувачів аніж аналізу контенту.

За підсумками детального аналізу механізмів та різновидів колаборативної фільтрації виявлено, що для побудови кореляції між користувачами найбільш доречно застосовувати user-based підхід та будувати кореляцію за допомогою косинусної міри.

3 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

Перш ніж переходити безпосередньо до опису реалізації ресурсу необхідно визначити специфікацію вимог до програмного засобу. Вона проектується для опису програмного продукту – який надає користувачу детальну інформацію стосовно готелів та кімнат, має широкі можливості пошуку з застосуванням методів колаборативної фільтрації а також дозволяє оформляти резервування та відслідковувати статус свого замовлення. Крім того специфікація вимог до програмного засобу описує системні, функціональні і не функціональні вимоги до даного продукту [21].

Головним призначенням специфікації вимог до програмного засобу є фіксування усіх вимог до продукту, а також опис поведінки майбутньої системи.

Область застосування створюваної системи – сфера обслуговування в туристичній галузі. Саме тому, потенційна аудиторія, яка буде користуватися ресурсом – туристи. Межі проекту визначають безпосередньо процесами які можливо автоматизувати. В рамках проекту, на створеному ресурсі клієнт матиме змогу ознайомитися з каталогом та зробити резервування, задля того щоб пришвидшити процес пошуку найбільш відповідного номеру він також матиме змогу користуватися фільтрами та системою рекомендацій. Адміністратор повинен мати можливість переглядати список усіх нових замовлень, та змінювати статус того чи іншого замовлення. Клієнт же, в свою чергу, повинен мати змогу відслідковувати зміни статусу свого замовлення, якщо він зареєстрований. Такі етапи як оплата вартості, заселення, проживання, виселення розглядатися не будуть.

Призначення продукту - автоматизація процесу надання клієнту детальної інформації стосовно готелів, а також резервування номерів та відслідковування статусу замовлення.

Перспективи впровадження продукту: даний продукт буде корисний як великим, так і малим компаніям, які надають послуги оренди онлайн. Враховуючий сучасний рівень розвитку Інтернет технологій продукт буде затребуваний. Більш того, готель який не має можливості по резервуванню онлайн ризикує втратити левову частину клієнтів. Користувачі продукту: туристи, люди які приїхали в місто по роботі.

Вимоги: система являє собою веб-орієнтований додаток, написаний на мові програмування Java з використанням технології Spring. В якості СУБД була вибрана MySQL. Так як продукт являє собою крос платформне рішення, його застосування не прив'язується до конкретної операційної системи. Потрібна налаштована мережа Internet і браузер. Для технічної підтримки проекту додатково потрібна налаштована віртуальна машина Java і контейнер сервлетів Tomcat.

Вимоги до розробки цього продукту:

- Середовище розробки: IDEA і сервер Tomcat;
- Необхідна встановлена СУБД – MySQL;
- Локальна мережа.

Розглянемо з чого складається робота системи. По-перше, програмний продукт повинен забезпечувати роботу двом категоріям користувачів: адміністраторам та клієнтам. В рамках цього, ідентифікація користувачів в відповідності з категорією повинна здійснюватися за допомогою логіна та паролю. При їх коректному вводі здійснюється перехід на робочу сторінку користувача, відповідно до введених даних. При не вірному вводі даних з'являється повідомлення про знайдену помилку. Також, в рамках цього пункту відбувається реєстрація. У випадку якщо користувач залишив деякі поля порожніми, або ж ввів інформацію, котра вже міститься в базі даних також з'являється повідомлення про помилку. В разі успіху, користувач переходить на сторінку де йому необхідно пройти процес ідентифікації.

По-друге, клієнт повинен мати змогу сформувати замовлення на резервування. Після заповнення форми з вибором дат заселення та виїзду і типу номеру потенційний клієнт повинен мати змогу перейти на сторінку усіх вільних номерів на вибрані дати та зазначеного типу. Користувач повинен мати можливість ознайомитися з детальним описом, дізнатися ціну та переглянути фото. Клієнт повинен мати змогу формувати список номерів готелів, котрі йому сподобались. Завдяки реалізованому алгоритму колаборативної фільтрації клієнт повинен мати змогу скористуватись функцією системи щодо знаходження найбільш відповідного його запитам номеру. Також йому повинен надаватися такий функціонал як додавання до кошику та видалення з кошику. Коли кошик сформований, користувач повинен мати можливість відправити замовлення адміністратору на обробку.

Третій етап представляє той функціонал системи, який надається адміністратору. Він повинен мати змогу передивлятися список замовлень та редагувати статуси замовлень. Система, що розробляється повинна бути простою в експлуатації.

Варто виділити також вимоги надійності. Перша з них встановлює, що середня тривалість часу між двома послідовними проявами помилок в системі повинна складати п'ять годин. Друга визначає допустиму ймовірність виходу системи з ладу. Ця ймовірність в нашому випадку не повинна перевищувати три відсотки. Остання вимога встановлює середній час відновлення системи – одна година.

3.1 Розробка структури БД

Процес створення схеми бази даних і визначення необхідних обмежень цілісності отримав назву – проектування баз даних. Це достатньо складний процес, від якого залежить коректність роботи усієї інформаційної системи. В ході

проектування баз даних необхідно вирішити задачі забезпечення зберігання в базі даних усієї необхідної інформації о системі, забезпечити можливість отримання даних по всім необхідним запитам, скорочення надмірності і дублювання даних, забезпечення цілісності даних. Виділимо основні етапи проектування баз даних.

1. концептуальне проектування – побудова семантичної моделі предметної області, тобто інформаційної моделі найбільш високого рівня абстракції. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних.

2. логічне проектування – створення схеми бази даних на основі конкретної моделі даних, наприклад, реляційної моделі даних. На етапі логічного проектування враховується специфіка конкретної моделі даних.

3. фізичне проектування - створення схеми бази даних для конкретної СУБД. Специфіка конкретної СУБД може включати в себе обмеження на іменування об'єктів бази даних, обмеження на підтримувані типи даних і тому подібне. Крім того, специфіка конкретної СУБД при фізичному проектуванні включає вибір рішень, пов'язаних з фізичним середовищем зберігання даних (вибір методів управління дисковою пам'яттю, поділ бази даних по файлам і пристроям, методів доступу до даних), створення індексів і так далі.

В базі даних, що розробляється в рамках виконання роботи необхідно зберігати інформацію стосовно: номерів готелів, клієнтів та замовлень на резервування. На першому кроці створення бази даних необхідно виділити ключові сутності і позначити зв'язки, які можуть бути встановлені між цими сутностями. Для цього буде використаний один з найвідоміших засобів опису концептуальних схем предметної області – модель «сутність-зв'язок». Вона являє собою формальну конструкцію, котра сама по собі не визначає ніяких графічних засобів її візуалізації. Саме тому, в якості графічної нотації була використана діаграма сутність-зв'язок.

В рамках виконання роботи для побудови концептуальної схеми моделі використана методологія IDEF1X, бо, як вже було зазначено раніше, в роботі використовується саме реляційна модель даних. На рисунку 3.1 представлена

логічна структура БД, побудована за допомогою IDEF1X-методології, яка відображає логічну структуру інформації щодо об'єктів системи [22].

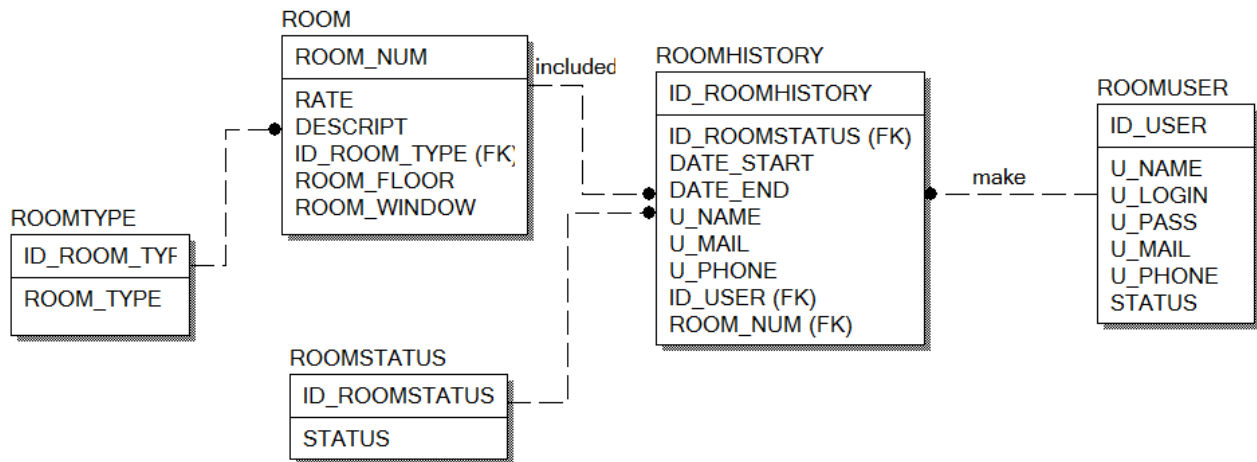


Рисунок 3.1 – Логічна модель

Сутність «ROOMTYPE» містить інформацію стосовно можливих типів житла. Вже на цьому етапі варто зазначити, що кількість видів обмежена і після вибору СУБД при побудові фізичної моделі на поле «ROOMTYPE» буде накладене обмеження на можливі варіанти значення цього атрибута. Поле «ID_ROOM_TYPE» унікально ідентифікує кожний з типів.

Сутність «ROOM» містить детальну інформацію стосовно номерів готелів. Задля спрощення проектування та спрощення системи в цілому будуть розглядатися окремі номери готелів, тобто один готель може мати в системі декілька номерів в залежності від типу номеру. Це зроблено для того аби не зробити проектування занадто складним. До цієї інформації відносять ID типу номеру, яке в цьому відношенні виступає зовнішнім ключем, детальний опис, ціну, поверх, бік на який виходять вікна та фото. В якості первинного ключа, що унікально ідентифікує кожний з екземплярів сутності виступає номер.

Сутність «ROOMSTATUS» містить інформацію стосовно можливих статусів замовлення. Вже на цьому етапі варто зазначити, що кількість видів статусу замовлення номеру обмежена і після вибору СУБД при побудові фізичної моделі на поле «STATUS» буде накладене обмеження на можливі варіанти значення цього атрибута. Поле «ID_ROOMSTATUS» унікально ідентифікує кожний зі статусів замовлення.

Сутність «ROOMUSER» містить детальну інформацію стосовно зареєстрованих клієнтів. До цієї інформації відносять ФІО клієнта, його поштову скриньку та контактний телефон, а також його логін з хешем паролю та статус, який в клієнта завжди дорівнює одиниці, а в адміністратора нулю. В якості первинного ключа, що унікально ідентифікує кожний з екземплярів сутності виступає атрибут «ID_USER».

Сутність «ROOMHISTORY» містить детальну інформацію стосовно замовлень на резервування номерів готелів. До цієї інформації відносять ФІО клієнта, його поштову скриньку та контактний телефон, дату заселення та дату виїзду. Атрибути «ID_ROOMSTATUS» і «ID_ROOMNUM», виступають зовнішніми ключами та містить інформацію щодо статусу замовлення та інформацію стосовно номеру відповідно. В разі, якщо клієнт зареєстрований на ресурсі інформація про нього надається завдяки зовнішньому ключу «ID_USER». В якості первинного ключа, що унікально ідентифікує кожний з екземплярів сутності виступає атрибут «ID_ROOMHISTORY».

3.2 Проектування програмного забезпечення

Для отримання графічного представлення логічної структури досліджуваної предметної області необхідно побудувати модель предметної області.

Логічна модель предметної області ілюструє сутності, а також їх взаємовідносини між собою. Сутності описують об'єкти, які є предметом діяльності предметної області і суб'єкти, що здійснюють діяльність в рамках предметної області. Властивості об'єктів і суб'єктів реального світу описуються за допомогою атрибутів. Взаємовідносини між сутностями ілюструються за допомогою зв'язків. Правила та обмеження взаємовідносин описуються за допомогою властивостей зв'язків. Зазвичай зв'язки визначають або залежності між сутностями, або вплив однієї сутності на іншу.

Побудована модель предметної області (рисунок 3.2) складається з наступних сутностей, що описані нижче.

Customer. Ця сутність описує одного із суб'єктів розглянутої предметної області - клієнта. У кожного користувача є атрибути - ім'я, телефон і пошта. Користувач створює замовлення і описує заявку на бронювання. Відноситься до категорії людей.

Order. Це об'єкт предметної області, що являє собою поточне замовлення (кошик), який створює користувач, в якому міститься список обраних ним на резервування номерів готелів. Відноситься до категорії контейнерів.

BookLineRoom. Це об'єкт предметної області, що являє собою список обраних номерів готелів, що міститься в кошику. Номери зі списку описуються з іншого об'єкта предметної області - опису номерів. BookLineRoom зберігає в собі також дати заселення і виселення клієнта. Відноситься до категорії вмісту контейнерів.

RoomSpecification - об'єкт предметної області, який зберігає в собі характеристики (номер, ціну, фото кімнати, опис кімнати). Він описує BookLineRoom і міститься в RoomCatalog. Відноситься до категорії опису об'єктів. В той час як RoomCatalog цю об'єкт предметної області, що являє собою каталог, доступних для резервування. Він формує RoomHistory. Відноситься до категорії каталогів.

Варто зазначити що RoomHistory це об'єкт предметної області, сформований за допомогою каталогу та зберігає в собі історію резервування. Відноситься до категорії каталогів.

Booking - це об'єкт предметної області, що є результатом реєстрації в системі замовлення, який в подальшому зберігається в RoomHistory, а описується за допомогою RoomSpecification і користувача. Відноситься до категорії транзакцій.

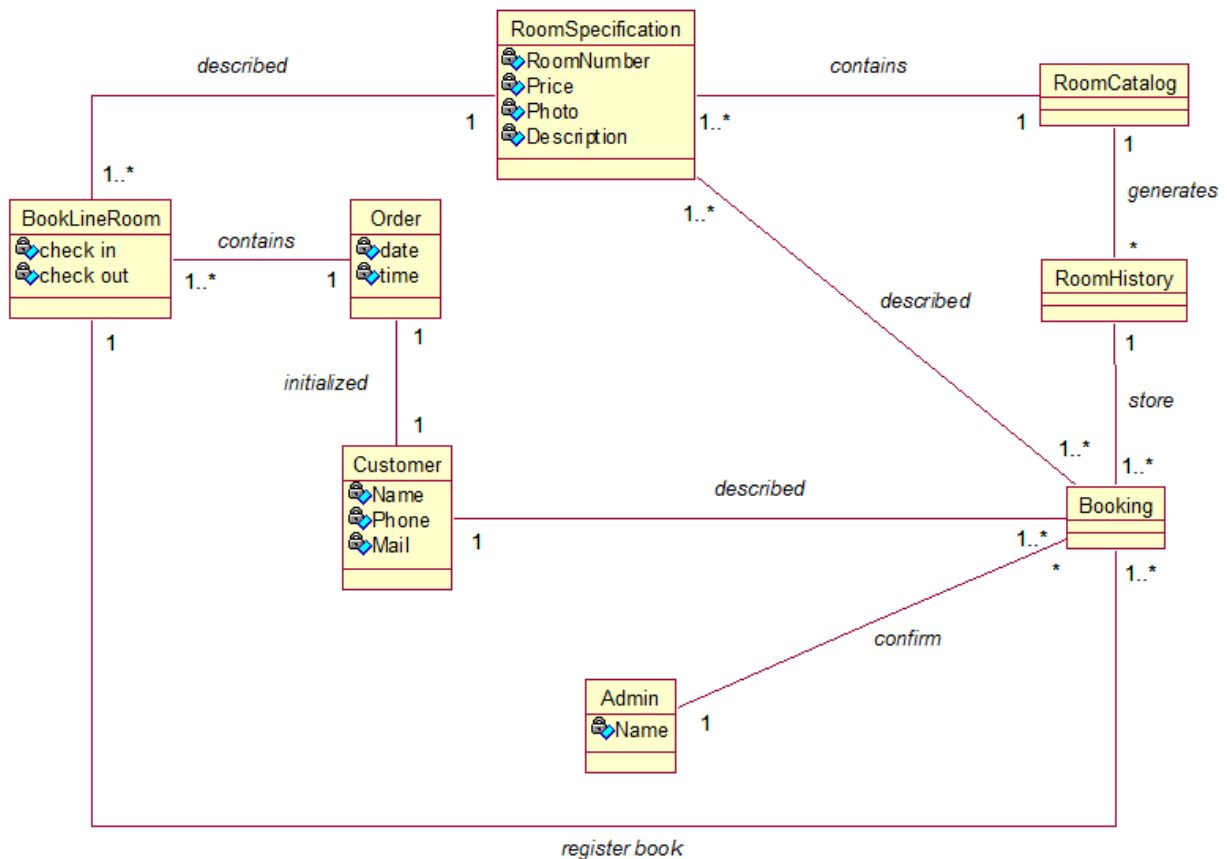


Рисунок 3.2 – Діаграма класів туристичної установи

Перед початком кодування була розроблена структура класів та їх відносин (рисунок 3.3). В UML діаграма класів описує статичну структуру системи, тобто визначає типи об'єктів системи і різного роду статичні зв'язки і відносини між ними. На діаграмах класів також зображуються атрибути класів, операції класів та обмеження, які накладаються на зв'язки між об'єктами.

Графічно класи представляють таким чином: верхня секція містить ім'я класу і інші загальні властивості. У середній секції міститься список атрибутів, а в нижній - список операцій. Атрибути зберігають інкапсульовані дані класу, а також визначають склад і структуру даних, що зберігаються в об'єктах цього класу. Операції описують поведінку об'єктів класу.

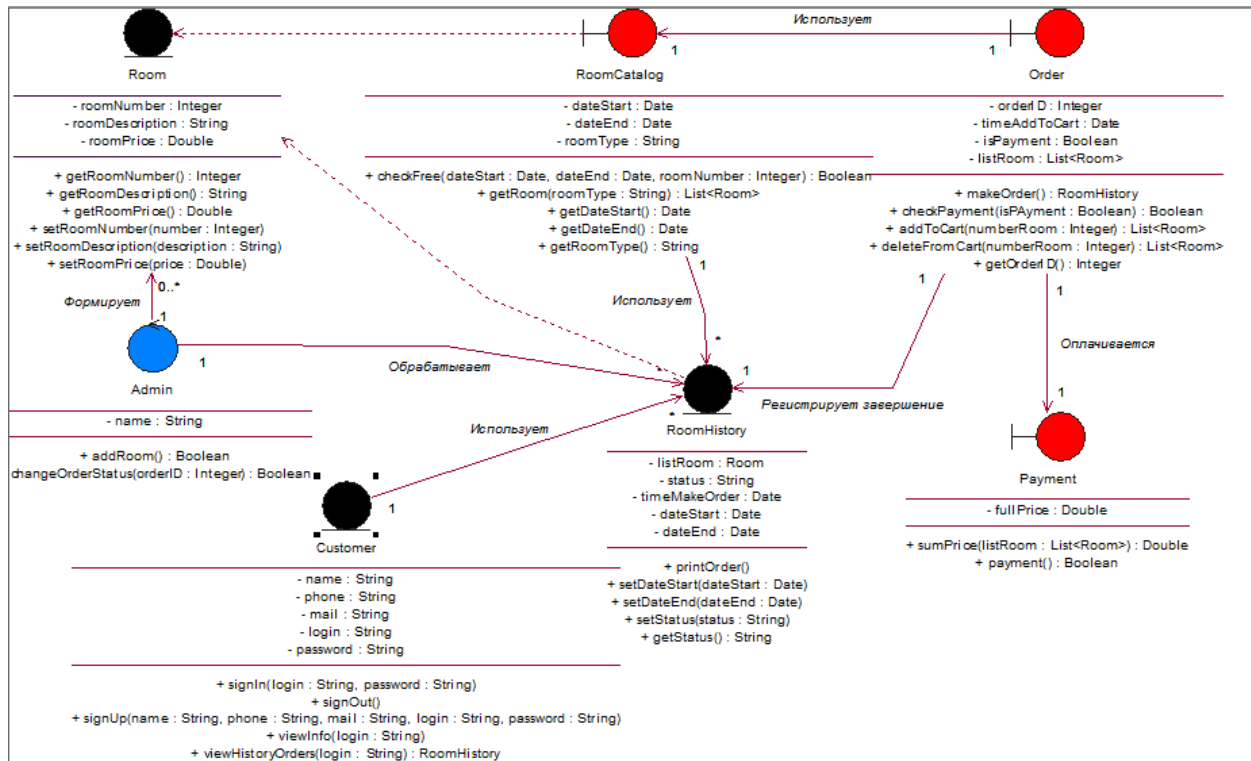


Рисунок 3.3 – UML-діаграма класів

Клас Room - пасивний клас, інформація про який повинна зберігатися постійно і не знищуватися з виключенням системи. Цей клас відповідає окремій таблиці в БД. Тому в якості стереотипу даного класу обраний стереотип <entity>. Атрибутами класу обрані поля, необхідні для опису кожного окремого взятого номера готелю - номер кімнати готелю, її опис та ціна за день. Методи даного класу - це методи, що дозволяють отримати інформацію про певний номер або встановити цю інформацію.

Клас Customer також є класом сутністю, так як він відповідає окремій таблиці в БД, що зберігає інформацію про всіх користувачів. В якості властивостей даного класу виступають - ім'я користувача, його телефон, пошта, логін і пароль. Методи даного класу вирішують питання реєстрації, авторизації та перегляду своєї особистої інформації і історії замовлень.

Клас RoomCatalog розташовується на кордоні системи з зовнішнім середовищем і безпосередньо взаємодіє з акторами, але при цьому є складовою частиною системи. Тому в якості стереотипу даного класу обраний стереотип - <boundary>. Атрибути даного класу - це поля, які користувач заповнює на головній сторінці сайту, для того щоб сформувати каталог. Залежно від значення цих полів отримують список всіх номерів готелів, обраного користувачем типу, а також перевіряється чи вільний номер на зазначені дати.

Клас Order є граничним класом, представляє з себе кошик, який містить вибрані користувачем номери готелів з каталогу. Поля цього класу - генерований orderID, необхідний для ідентифікації замовлення, час додавання в кошик номеру і прапор, який відображує стан оплати. Ці атрибути необхідні для перевірки була проведена оплата і якщо відповідь негативна, то після закінчення півгодини резервування з номера автоматично знімається. Серед методів цього класу - оформлення замовлення, перевірка на оплату, додавання в кошик і видалення з кошика.

Клас Payment є граничним класом, бо являє з себе інтерфейс з методом оплати. Його єдиний атрибут - це підрахована за допомогою методу sumPrice () ціна бронювання. Другий метод Payment відповідає безпосередньо за функцію оплати.

Взаємодії між об'єктами в системі представляються діаграмами взаємодії. Діаграми взаємодії поділяються на два основних типи діаграм: діаграми послідовності і кооперативні діаграми.

Діаграми послідовності і кооперативні діаграми несуть в собі одну інформацію, але виражену різними способами. Діаграми послідовності показують

взаємодію об'єктів в часі і відображають послідовність подій, що відбуваються. Проте, на діаграмі не відбиваються зв'язки між об'єктами. Кооперативні діаграми дозволяють просторово розташовувати об'єкти, для того щоб краще уявити взаємодію між об'єктами. Тимчасова послідовність переданих повідомлень відбивається за допомогою нумерації повідомлень. На рисунку 3.4 наведена діаграма послідовності для прецеденту - оформлення резервування.

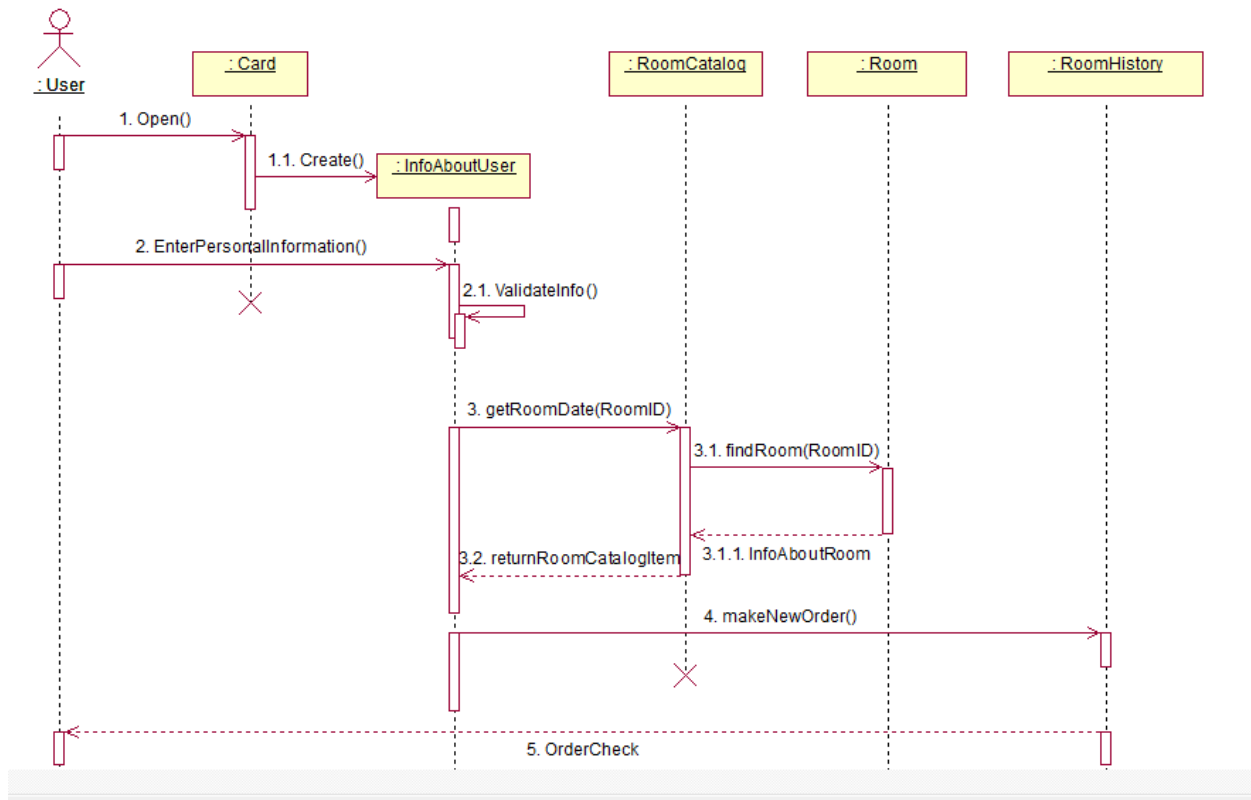


Рисунок 3.4 - Діаграма послідовності

На діаграмі кооперації зображується взаємодія між частинами системи або ролями кооперації.

Відмінність діаграми кооперації від діаграми послідовностей полягає в тому, що на діаграмі кооперації явно вказуються відносини між об'єктами, а час не використовується як окремий вимір (замість цього вказують порядкові номери

виклику об'єктів). На рисунку 3.5 побудовано діаграму кооперації для прецеденту - оформлення резервування.

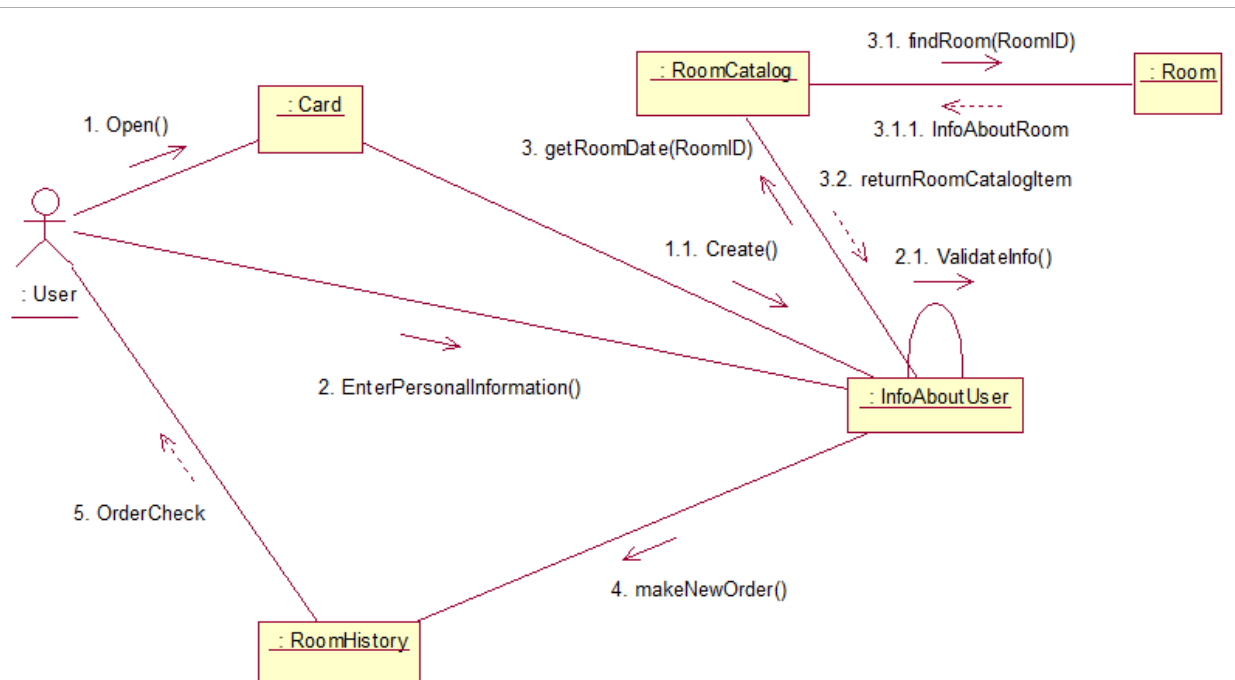


Рисунок 3.5 - Діаграма кооперації

Об'єкти характеризуються поведінкою і станом, в якому знаходяться. Діаграми станів застосовуються для того, щоб пояснити, яким чином працюють складні об'єкти.

Стан - ситуація в життєвому циклі об'єкта, під час якої він задовольняє деякій умові, виконує певну діяльність або очікує якоїсь події. Стан об'єкта визначається значеннями деяких його атрибутів і присутністю або відсутністю зв'язків з іншими об'єктами.

Діаграма станів показує, як об'єкт переходить з одного стану в інший. Очевидно, що діаграми станів служать для моделювання динамічних аспектів системи. Діаграма станів використовується при моделюванні життєвого циклу об'єкта. Для системи, що проектується, було побудовано діаграму станів (див. рис.

3.6), що відображає загальний алгоритм переходу замовлення на резервування з одного стану до іншого [23].

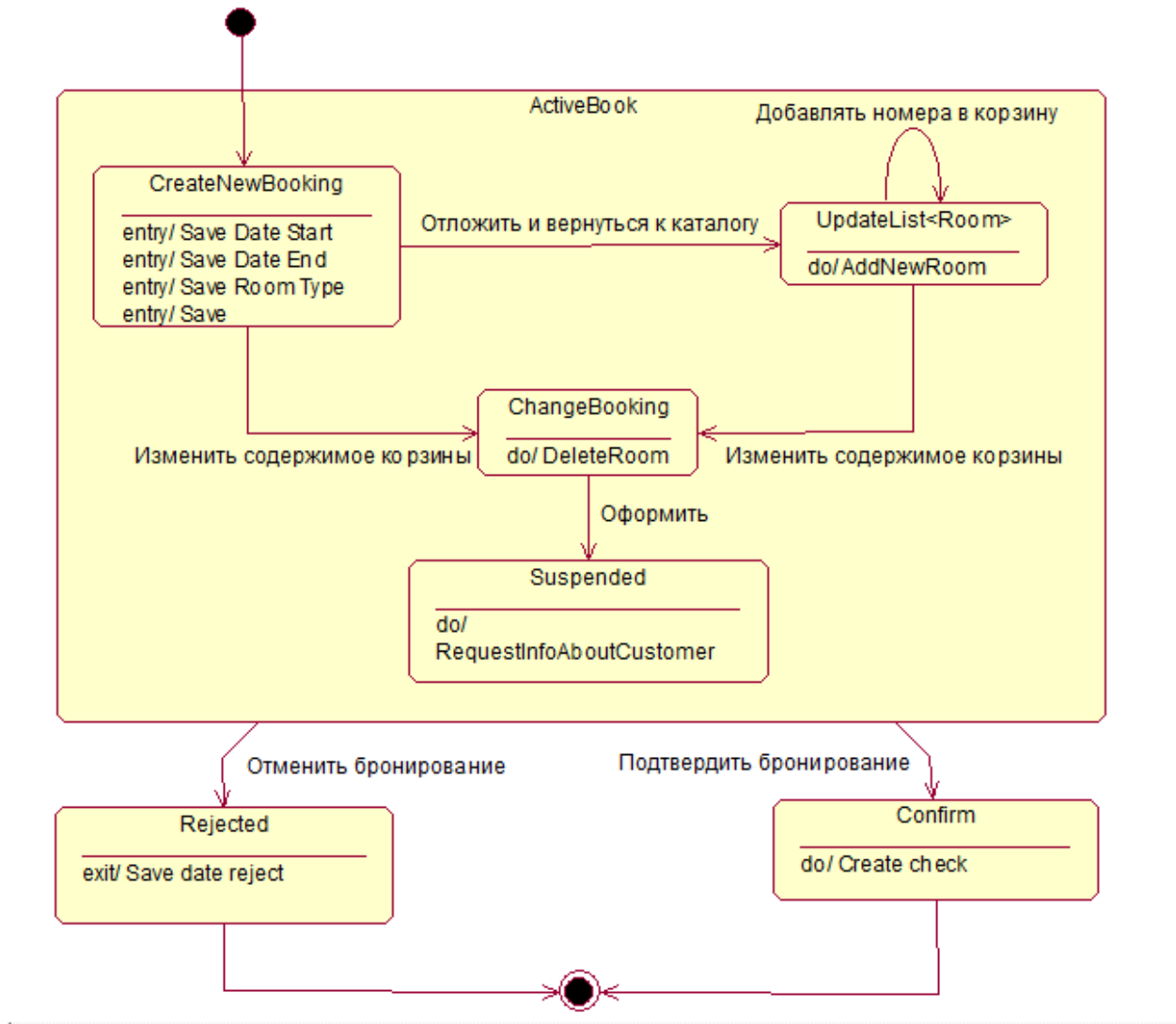


Рисунок 3.6 - Діаграма станів

3.3 Опис обраних технологій та демонстрація

В роботі для роботи з даними була використана СУБД MySQL. Сервер MySQL забезпечує ефективні і дієві рішення для основних засобів баз даних.

Коротко розглянемо кожну з них. Сервер MySQL підтримує найбільші бази даних - потенційним розміром до сотень гігабайт. Щоб забезпечити дієвий контроль за використанням дорогих дискових пристроїв, він надає повний контроль розподілу простору. Також він підтримує велику кількість користувачів, що одночасно виконують різноманітні додатки, які оперують одними і тими ж даними. Він мінімізує суперництво за дані і гарантує узгодженість даних. Сервер MySQL підтримує всі описані вище можливості, при цьому зберігаючи високу ступінь сумарної продуктивності системи. Щоб витягти максимум переваг з наявної комп'ютерної системи або мережі, MySQL дозволяє розділяти роботу між сервером бази даних і прикладними програмами клієнтів.

Як мова розробки була обрана Java - об'єктно-орієнтована мова програмування. Програми написані на Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині незалежно від комп'ютерної архітектури.

Робота на різних платформах гетерогенних мереж відкидає традиційну схему розподілу програмного забезпечення, версій ПЗ, модифікації ПЗ, об'єднання ПЗ і так далі. Java здатна вирішувати проблеми гетерогенних середовищ завдяки: нейтральності до архітектури та змозі динамічно підстроюватися.

В якості ORM-рішення для мови Java, в роботі використовується технологія Hibernate, яка не тільки дбає про зв'язок Java класів з таблицями бази даних (і типів даних Java в типи даних SQL), але також надає можливості для автоматичної побудови запитів і отримання даних і може значно зменшити час розробки, який зазвичай витрачається на ручне написання SQL і JDBC коду. Hibernate генерує SQL виклики і звільняє розробника від ручної обробки результуючого набору даних і конвертації об'єктів, зберігаючи програму трансформацію в усі SQL бази даних.

Також в програмній реалізації використовується Spring Framework, який забезпечує комплексну модель розробки і конфігурації для сучасних бізнес-додатків на Java - на будь-яких платформах. Ключовий елемент Spring - підтримка

інфраструктури на рівні програми: основна увага приділяється "водопроводу" бізнес-додатків, тому розробники можуть зосередитися на бізнес-логіці без зайвих налаштувань в залежності від середовища виконання.

Клієнтський інтерфейс розроблений на наступних технологіях: HTML для формування структури сторінки, CSS для оформлення сформованих сторінок, Bootstrap для оформлення сформованих сторінок в спеціальному стилі а JavaScript для валідації введених користувачем даних та анімації.

Для імплементації алгоритму колаборативної фільтрації була використана бібліотека Apache Mahout. Це потужна, масштабована бібліотека машинного навчання, яка працює на вершині Hadoop MapReduce. Машинне навчання - це дисципліна штучного інтелекту, яка дозволяє системам навчатися на основі даних, постійно підвищуючи продуктивність, оскільки обробляється більше даних. Машинне навчання - це основа багатьох технологій, які є частиною нашого повсякденного життя.

Mahout пропонує специфічний набір досить детальних функціональних можливостей для машинного навчання на великих даних. Mahout - це, по суті, середовище або структура, яка забезпечує будівельні блоки для визначення та виконання алгоритмів машинного навчання, які можуть обробляти дані в масштабі за допомогою дуже простого програмування.

В контексті предметної області що розглядається, маємо оцінки користувачів для готелів (рис. 3.7):

1	1,10,1.0
2	1,11,2.0
3	1,12,5.0
4	1,13,5.0
5	1,14,5.0
6	1,15,4.0
7	1,16,5.0
8	1,17,1.0
9	1,18,5.0
..	- - - - -

Рисунок 3.7 - Оцінки користувачів для готелів

Для знаходження рекомендації маємо лише викликати декілька методів з заданими параметрами як показано на рис. 3.8.

```
UserNeighborhood neighborhood = new ThresholdUserNeighborhood(0.1,similarity, model);
UserBasedRecommender recommender = new GenericUserBasedRecommender(model, neighborhood, similarity);

// The First argument is the userID and the Second parameter is 'HOW MANY'
List<RecommendedItem> recommendations = recommender.recommend(2, 2);
```

Рисунок 3.8 - Рекомендація за Apache Mahout

За результатами роботи рекомендаційного механізму отримаємо ідентифікатори елементів які найімовірніше сподобаються користувачу. Для порівняння на рисунку 3.9 зображено звичайний запит до бази даних, а на рисунку 3.10 з урахуванням результатів роботи рекомендаційного алгоритму.

```

4 • SELECT * FROM room WHERE id_roomtype = 1
5 AND room_num NOT IN (SELECT rh.room_num from roomhistory as rh where date_start < 05-22-2020 and date_end > 05-26-2020 and rh.id_roomstatus <> 3);

```

room_num	id_roomtype	rate	descript	src_photo
201	1	30	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
202	1	35	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
203	1	31	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
204	1	40	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
205	1	50	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
206	1	24	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
207	1	31	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
208	1	43	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
209	1	28	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
210	1	10	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
211	1	39	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
212	1	44	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
213	1	48	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
214	1	100	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
215	1	29	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
216	1	31	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
217	1	80	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
218	1	77	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
219	1	64	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg

Рисунок 3.9 – Запит без урахування рекомендацій

```

4 • SELECT * FROM room WHERE id_roomtype = 1
5 AND room_num NOT IN (SELECT rh.room_num from roomhistory as rh where date_start < 05-22-2020 and date_end > 05-26-2020 and rh.id_roomstatus <> 3)
6 AND room_num IN (204, 207);

```

room_num	id_roomtype	rate	descript	src_photo
204	1	40	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
207	1	31	The rooms in CaPisani hotel have been realised ...	D:KnowHow.jpg
NULL	NULL	NULL	NULL	NULL

Рисунок 3.10 – Запит з урахуванням рекомендацій

Завдяки механізму колаборативної фільтрації користувач має змогу отримати не повний список номерів а тільки ті, які йому найімовірніше сподобаються. Це дозволить заощаджувати велику кількість часу користувачам системи а власникам утримувати клієнтів на ресурсі та допомагати їм прийняти рішення в разі того якщо вони матимуть сумніви стосовно кращого вибору.

Переходячи до інтерфейсу користувач системи зверне свою увагу на вікно з формою бронювання , воно містить бажані дати заселення та виселення, а також вибраний тип. Знайти цю форму достатньо легко, бо вона розташована на головній сторінці ресурсу. На рисунку 3.11 представлений графічний інтерфейс цієї форми.

The image shows a digital reservation form overlaid on a background of a person in a suit. The form has a blue header with the text 'MAKE A RESERVATION'. Below the header, there are three input fields: 'Check in:' with the value '23.06.2018', 'Check out:' with the value '26.06.2018', and 'Type:' with a dropdown menu showing 'Standart'. At the bottom of the form is a white button with the text 'Check availability'.

Рисунок 3.11 – Форма бронювання

Для того, щоб функція коректно відпрацьовувала при всіх можливих діях користувача, потрібна валідація даних. В рамках реалізації компоненту була розроблена валідація даних і на клієнті, з використанням регулярних виразів мови JavaScript і стандартних тегів HTML5, і на сервері, з використанням конструкцій мови програмування Java. Саме такий підхід є доречним, так як в разі збою одного валідатора, завжди є другий, який проконтролює правильність введених клієнтом даних.

В результаті, в разі, якщо користувач введе некоректні дані сервіс не впаде, не перестане працювати і не стане видавати невірну інформацію, а просто пере направить користувача знову на форму введення з проханням відкоригувати введену раніше інформацію. Алгоритм роботи сервісу з урахуванням валідації даних представлений на рисунку 3.12.

Робота практично усіх функцій системи має достатньо схожий алгоритм, від користувача надається певна вхідна інформація, яка потім валідується, в разі успіху відбувається запит до БД за результатами якого користувачу відображається та чи інша інформація, тож інші алгоритми системи не будуть описані детально.

Так як забронювати номери готелів можна і не зареєструвавшись в системі, то алгоритм роботи методу бронювання має дві гілки, в разі якщо користувач пройшов авторизацію в системі йому достатньо лише обрати бажаний номер та натиснути кнопку ‘замовити’, але якщо він невідомий для системи йому буде запропоновано ввести додаткову особисту інформацію перед оформленням замовлення. Ця форма представлена на рисунку 3.13.

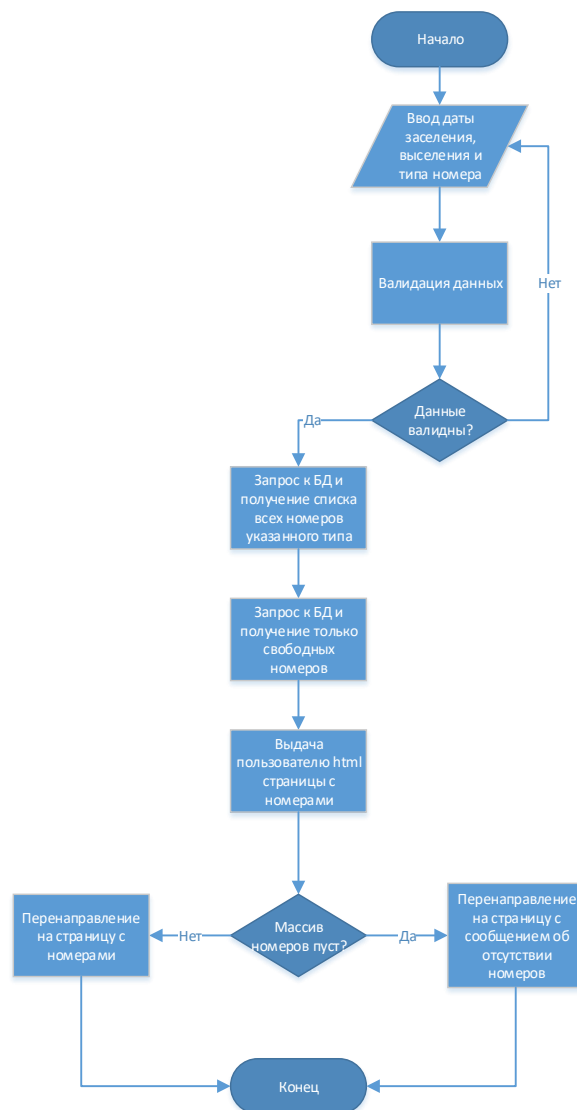


Рисунок 3.12 – Алгоритм роботи сервісу з урахуванням валідації

Please enter your personal details



The form consists of three input fields stacked vertically on a dark background. The first field is labeled 'Full Name:' and contains the placeholder text 'Full Name'. The second field is labeled 'Mail:' and contains the placeholder text 'Mail'. The third field is labeled 'Phone:' and contains the placeholder text 'Phone'. Below the fields is a blue button with the text 'Make order' and a small icon of a document with a pencil.

Рисунок 3.13 – Форма вводу додаткової інформації

Після оформлення резервування користувачу надається уся необхідна інформація, представлена у вигляді таблиці. Окрім цього, на зазначену поштову скриньку надсилається письмо, що містить повну інформацію стосовно замовлення і інструкції по сплаті та контактна інформація. Вигляд готового замовлення зображений на рисунку 3.14.

Your order

USER NAME	NUMBER	DATA START	DATA END	PRICE
Danil Bugay	103	14.06.2016	14.06.2016	200

Print 

Рисунок 3.14 – Відображення готового замовлення

Адміністратор ресурсу має змогу переглядати усі нові замовлення та змінювати їх статус. Для зручності інформація стосовно замовлень також

представлена у вигляді таблиці, де кожний новий рядок представляє з себе нове замовлення. Вигляд таблиці з заявами на резервування зображений на рисунку 3.15.

Please select the hotel room

NUMBER	DATA START	DATA END	PRICE	USER NAME	USER PHONE	SELECT
102	29.05.2018	31.05.2018	200.0	Danil Bugay	+380985804515	<input type="checkbox"/>

Рисунок 3.15 – Відображення нових заяв на резервування

В свою чергу клієнт, в разі якщо він зареєстрований в системі, має змогу на своїй домашній сторінці переглядати історію замовлень, тим самим відслідковуючи статус заявки, як показано на рисунку 3.16.

History of orders

NUMBER	DATA START	DATA END	PRICE	STATUS
101	29.05.2018	31.05.2018	200.0	Application denied
102	29.05.2018	31.05.2018	200.0	Application denied

Рисунок 3.16 – Відображення історії замовлень

Також ресурс має рекомендаційних механізм заснований на алгоритмі колаборативної фільтрації, описаний раніше. Інтерфейс форми пошуку зображений на рисунку 3.17

Price: Windows side: Floor:

Рис. 3.17 – Інтерфейс форми пошуку

Після того як клієнт натисне кнопку знайти найкращий номер, на сторінці каталогу з'явиться та спеціальна пропозиція, яка найімовірніше сподобається користувачу (рисунок 3.18).

Price: Windows side: Floor:


NUMBER	DESCRIPTION	PRICE	PHOTO	SELECT
101	This category is always subject to interpretation. It is supposed to mean superior to a standard room in both size and furnishings, but it often refers to just the view. Some hotels have only Superior rooms; the categories then are defined by the view and location of the room. Windows side - east/sea. Floor - 4	240.0		<input type="button" value="Add to cart"/>

Рисунок 3.18 – Результат пошуку номеру

3.4 Висновки за розділом 3

В розділі були описані та розроблені програмні рішення та проілюстровані результати реалізації алгоритму колаборативної фільтрації в ІС туристичної установи. Практичні дослідження підтвердили ефективність впровадження рекомендаційних механізмів до інформаційних систем, та можуть слугувати гарною основою для прийняття рішення щодо інтеграції рекомендаційних алгоритмів до існуючих інформаційних ресурсів туристичних установ.

ВИСНОВОК

В атестаційній роботі була вирішена задача впровадження рекомендаційного алгоритму в інформаційну систему для туристичної установи, яка займається пошуком і бронюванням готелів задля того щоб здійснювати підбір найбільш відповідного готелю для користувача, на основі його історії бронювання і кореляції з іншими користувачами. Це дозволило заощаджувати велику кількість часу користувачам системи а власникам утримувати клієнтів на ресурсі та допомагати їм прийняти рішення в разі того якщо вони матимуть сумніви стосовно кращого вибору.

Для вирішення цієї задачі був проведений аналіз існуючих рекомендаційних алгоритмів та виявлено що колаборативна фільтрація що базується на кореляції користувачів найкраще підходить для даної предметної області а міра косинуса найбільш доречна для побудови кореляції між користувачами.

Для того аби це зробити була детально проаналізована уся система з застосуванням різних методологій проектування та побудовані моделі предметної області що дозволили детально розглянути та зрозуміти процеси що відбуваються.

Як результат була отримана математична модель та побудована програма, а саме веб-ресурс для туристичної установи особливістю якого став розумний пошук що базується на імплементації алгоритму колаборативної фільтрації за допомогою бібліотеки Apache Mahout. Побудова показала доречність застосування рекомендаційних механізмів в інформаційних системах загалом, та зокрема в туристичній установі.

За темою атестаційної роботи було опубліковано тези до міжнародної конференції "Modern global trends in the development of innovative scientific researches" (Riga, Latvia, March 20, 2020).

ПЕРЕЛІК ПОСИЛАНЬ

1. Маклаков С.В. ВРwin и ERwin. CASE-засоби розробки інформаційних систем / С.В. Маклаков. - М.: Диалог-МИФИ, 1999. - 256 с.
2. Рамбо Дж., Блаха М. UML 2.0. Об'єктно-орієнтоване моделювання і розробка. - П.: Питер, 2007. – 544 с.
3. Вендров А. CASE-технології. Сучасні методи і засоби проектування інформаційних систем. - М.: Статистика, 1998 г. – 176 с.
4. Лешек А. Аналіз вимог і проектування систем. Розробка інформаційних систем з використанням UML. - Вільямс, 2002 г. – 432с.
5. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry” *Communications of ACM*, vol. 35, no. 12, pp. 61–70, 1992.
6. G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
7. Федоровский, А.Н. Архитектура рекомендательной системы, работающей на основе неявных пользовательских оценок / А.Н. Федоровский, В.К. Логачева // - RCDL'2011, Воронеж, Россия, 2011
8. Правиков, А.А. Разработка и применение метода формализации проектирования рекомендательных систем с естественно-языковым интерфейсом: дис. канд. технических наук, Москва, 2011. С. 160
9. Нефедова, Ю. С. Архитектура гибридной рекомендательной системы GEFEST (*Generation-Expansion-Filtering-Sorting-Truncation*, Системы и средства информ.). 2012. Том 22. Вып 2. С. 176-196

10. Vasily, A. *Supervised Tag Extraction in a Web Recommender System, Similarity Search and Applications/* A. Vasily, S. Leksin, I. Nikolenko //Semi Lecture Notes in Computer Science Volume 8199. 2013. P. 206-212

11. Ignatov, D. I. *Online Recommender System for Radio Station Hosting: Experimental Results Revisited, in: Proceedings of The 2014 /S. Nikolenko, A. Таймураз, N. Konstantinova //IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2014, 11-14 August 2014 Warsaw, Poland /* Hayч. ред.: D. Slezak, H. S. Nguyen, M. Reformat, S. J. Eugene. Los Alamitos, California/Washington/Tokyo: IEEE Computer Society Conference Publishing Services (CPS), 2014. P. 229-236.

12. Goldberg, David. Douglas *Try Using collaborative filtering to weave an information Tapestry/* David Goldberg, David Nichols, Brian M. Oki, //Communication ofthe ACM, Dec. 1992. V/35. N. 12. P. 61-71.

13. Koren Y. *Matrix factorization techniques for recommender systems/* Y. Koren, R. M. Bell, C. Volinsky // Computer.V. 42. P. 30-37. 2009.

14. A. Ansari, S. Essegaiier, and R. Kohli, “*Internet recommendation systems*” *Journal of Marketing Research*, vol. 37, no. 3, pp. 363–375, 2000.

15. Shardanand U., Maes P. *Social Information Filtering: Algorithms for Automating «Word of Mouth»* // Proc. Conf. Human Factors in Computing Systems, 1995 . P. 210-217.

16. Goldberg D., Nichols D., Oki B. M., Terry D. *Using collaborative filtering to weave an information Tapestry* // Special issue on information filtering, 1992. Vol. 35, Issue 12 P. 61-70.

17. Adomavicius G., Tuzhilin A. *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions* // IEEE Transactions on Knowledge and Data Engineering, 2005. Vol. 17, Issue 6. P 734-749.

18. Celma O. *Music Recommendation and Discovery*. Springer-Verlag Berlin Heidelberg, 2010. 194 p.

19. Блинов И.Н., Романчик В.С., Java. *Методи програмування*. – М: Четыре четверти, 2013. – 896с.
20. Richardson L., Amundsen M. *RESTful Web APIs: Services for a Changing. OReily Media*, 2013. – 406 с.
21. Spurlock J. *Responsive Web Development*. OReily Media, 2013. – 128 с.
22. Гарсиа-Молина Г. *Системы баз данных*. Вильямс, 2003. – 1088 с.
23. Chapman N., Chapman J. *Authentication and Authorization on the Web*. 2012. – 246 с.
24. Методичні вказівки до організації виконання та захисту кваліфікаційної роботи ОКР «бакалавр» за напрямом 6.050101 – «Комп'ютерні науки» для студентів усіх форм навчання [Текст] / Упоряд.: М.В. Євланов, В.Г. Іванов, Л.М. Ребезюк, Н.В. Рябова. – Харків: ХНУРЕ, 2016. – 58 с.