

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

Навчально-науковий інститут комп'ютерних наук
та інформаційних технологій
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

Магістерська кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему: Розроблення структури сховища даних для продажу
автомобілів

Виконав: студент VI курсу, групи КН-62м
спеціальності

122 – “Комп'ютерні науки”
(шифр і назва напрямку підготовки, спеціальності)

Шалавило В.Т.
(прізвище та ініціали)

Керівник Борецька І.Б.
(прізвище та ініціали)

Рецензент Микошкобік Ю.Г.
(прізвище та ініціали)

Львів – 2024 р.

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій


Кафедра комп'ютерних наук

Рівень вищої освіти другий (магістерський)

Спеціальність 122 "Комп'ютерні науки"
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

 Борецька І. Б.

"05" січня 2024 року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Шалавило Володимир Тарасович
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення структури сховища даних для продажу автомобілів

керівник роботи, Борецька Ірина Богданівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 13.02.2023 року № С-49

2. Термін подання студентом роботи 05.01.2024 р.

3. Вихідні дані до роботи:

- провести огляд існуючих методів та засобів і визначити проблеми при пошуку рекомендацій структури сховища даних для продажу автомобілів;
- створити та дослідити алгоритм для пошуку рекомендацій структури сховища даних для продажу автомобілів;
- розробити архітектуру та реалізувати рекомендаційну систему вибору автомобілів на платформі Dynamics 365;
- провести тестування роботи розробленого програмного продукту.

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

Розділ 1. Стан проблемної області

Розділ 2. Інформаційне забезпечення

Розділ 3. Математичне забезпечення

Розділ 4. Програмне забезпечення

Розділ 5. Розроблення стартап-проекту

Висновки

5. Перелік графічного матеріалу:

системний аналіз, розробка та відображення алгоритму пошуку рекомендацій структури сховища даних для продажу автомобілів, структура програмного рішення, тестування

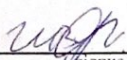
Додаток А. Вихідний код розробленого програмного забезпечення

6. Дата видачі завдання 15 лютого 2023 р.

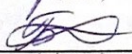
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Отримання завдання	15.02.2023	виконано
2	Огляд літературних джерел	30.04.2023	виконано
2	Розділ 1. Стан проблемної області	31.07.2023	виконано
3	Розділ 2. Інформаційне забезпечення	30.08..2023	виконано
4	Розділ 3. Математичне забезпечення	29.09.2023	виконано
5	Розділ 4. Програмне забезпечення	31.10.2023	виконано
7	Розділ 5. Розробка стартап-проекту	30.11.2023	виконано
8	Оформлення пояснювальної записки	29.12.2023	виконано
9	Подання готової роботи	05.01.2024	виконано

Студент


(підпис)Шалавило В.Т.
(прізвище та ініціали)

Керівник роботи


(підпис)Борецька І.Б.
(прізвище та ініціали)

АНОТАЦІЯ

Магістерська робота містить 88 сторінок пояснювальної записки, 43 рисунка, 21 таблиця, 1 додаток, 28 джерел.

Автодилери та компанії з продажу автомобілів отримують велику кількість запитів від клієнтів щодо придбання автомобілів. Тому співробітники відділу продажів отримують та аналізують побажання кожного клієнта та зазвичай особисто підбирають для клієнтів найкращі варіанти. Розроблено структуру зберігання даних з продажу автомобілів, що дозволяє спростити процес вибору та дати рекомендації, які не залежать від людського фактору та будуть більш оптимальними за критеріями замовника.

Ключові слова: *автомобіль, евклідова відстань, інтеграція, лексикографічна подібність, рекомендаційна система, CRM система.*

ANNOTATION

The master's thesis contains 88 pages of an explanatory note, 43 figures, 21 table, 1 appendix, 28 sources.

Car dealers and car sales companies receive a large number of inquiries from customers about purchasing cars. Therefore, the sales department employees receive and analyze the wishes of each customer and usually personally select the best options for customers. A structure for storing data on car sales has been developed, which allows you to simplify the selection process and give recommendations that do not depend on the human factor and will be more optimal according to the customer's criteria.

Keywords: *car, Euclidean distance, integration, lexicographic similarity, recommender system, CRM system.*

ТЕХНІЧНЕ ЗАВДАННЯ

Необхідно розробити програмне та алгоритмічне забезпечення структури сховища даних для продажу автомобілів, а саме:

1. проаналізувати існуючі методи та засоби і визначити проблеми при пошуку рекомендацій структури сховища даних для продажу автомобілів;
2. створити та дослідити алгоритм для пошуку рекомендацій структури сховища даних для продажу автомобілів;
3. розробити архітектуру рекомендаційної системи вибору автомобілів на платформі Dynamics 365;
4. реалізувати рекомендаційну систему вибору автомобілів на платформі Dynamics 365;
5. провести інтерпретацію отриманих результатів;
6. розробити програмне забезпечення для представлення результатів роботи.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ОДИНИЦЬ І ТЕРМІНІВ	8
ВСТУП.....	9
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ	11
1.1. Опис процесу діяльності.....	11
1.2. Актори і функції	12
1.3. Структурна схема діяльності з підбору рекомендацій	13
1.4. Постановка задачі.....	14
1.5. Рішення з інформаційного забезпечення	15
Висновки до розділу 1.....	19
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	20
2.1. Постановка задачі.....	20
2.2. Огляд методів розв’язання	20
2.3. Рекомендаційні системи	23
Висновки до розділу 2.....	31
РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	33
3.1. Розробка алгоритму пошуку рекомендацій	33
3.2. Приклад роботи алгоритму.....	37
3.3. Результати експериментальних досліджень	43
Висновки до розділу 3.....	45
РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	47
4.1 Засоби розробки.....	47
4.2 Архітектура програмного забезпечення.....	51
4.2.1. Структурна схема класів.....	51
4.2.2 Структурна схема послідовності	57
4.3. Інструкція користувача.....	57
4.3.1 Інструкція клієнта.....	57
4.3.2 Інструкція робітника компанії з продажу автомобілів	60
4.4. Опис технічного забезпечення	74
Висновки до розділу 4.....	74

	7
РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	75
5.1 Опис ідеї проєкту.....	75
5.2. Розроблення ринкової стратегії	80
5.3. Розроблення маркетингової програми	82
5.4. Вимоги до технічного та програмного забезпечення	85
Висновки до розділу 5.....	87
ВИСНОВКИ	88
СПИСОК ЛІТЕРАТУРИ	90
ДОДАТОК А. ЕКРАННІ ФОРМИ	94

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ОДИНИЦЬ І
ТЕРМІНІВ**

АРМ	автоматизоване робоче місце
ЕВ	евклідова відстань
ЛП	лексикографічна подібність
РС	рекомендаційна система
CRM	Customer Relationship Management

ВСТУП

Актуальність дослідження. Сучасний світовий автомобільний ринок є одним із важливих елементів світової економіки, одним із найбільш конкурентних ринків. Продажі автомобілів безпосередньо впливають на економічне зростання та кризові явища у світовій економіці[1].

Світовий авторинок у вересні цього року зріс на 0,5% до 7,647 млн легкових автомобілів [2].

У зв'язку з високим попитом на даний товар в автосалони і компанії з продажу автомобілів надходить велика кількість запитів від клієнтів про придбання автомобілів. Тому торговий персонал приймає і аналізує побажання кожного клієнта і, як правило, особисто підбирає для нього оптимальні варіанти.

Однак такий процес відбору займає багато часу і залежить від кваліфікації працівника, який знаходить рекомендацію, тобто отримані рекомендації залежать від людського фактору і не завжди можуть бути правильними. Тому важливим є обраний напрямок дослідження та створення системи рекомендацій.

Мета дослідження - підвищення ефективності роботи менеджерів, що в цілому збільшить прибуток компанії за рахунок збільшення загальної кількості оброблених заявок і продажів.

Об'єкт дослідження - процес пошуку списку рекомендованих автомобілів без урахування попередньої історії клієнта.

Предмет дослідження – методи та алгоритми пошуку рекомендацій

Для досягнення мети необхідно виконати такі **завдання**:

- проаналізувати існуючі методи та інструменти та виявити проблеми в пошуку рекомендацій;
- створити та протестувати алгоритм пошуку рекомендацій;
- розробити архітектуру системи рекомендацій на платформі Dynamics 365;

– впровадження системи рекомендацій на платформі Dynamics 365.

Наукова новизна одержаних результатів передбачає пошук рекомендацій на основі побажань клієнта, без будь-якої попередньої історії для конкретного клієнта.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Опис процесу діяльності

Перед покупкою автомобіля клієнт заповнює свої побажання щодо майбутнього автомобіля на сайті автопродавчої компанії або по телефону. Потім співробітники відділу продажів аналізують побажання кожного клієнта і підбирають для нього оптимальні варіанти. Потім записують варіанти в довільній формі. Таким чином, через відсутність автоматизації обробки інформації дані можуть бути втрачені і знадобиться новий пошук. Після знаходження рекомендації торговий співробітник інформує клієнта про можливі варіанти придбання автомобіля. Якщо після спілкування клієнт купує автомобіль, то торговий співробітник зобов'язаний видалити цей автомобіль з усіх джерел, щодо яких він знаходить рекомендації. Бізнес-процес обробки запиту клієнта на вибір автомобіля для впровадження системи рекомендацій наведено на рис. 1.1 за допомогою діаграми AS-IS IDEF0.

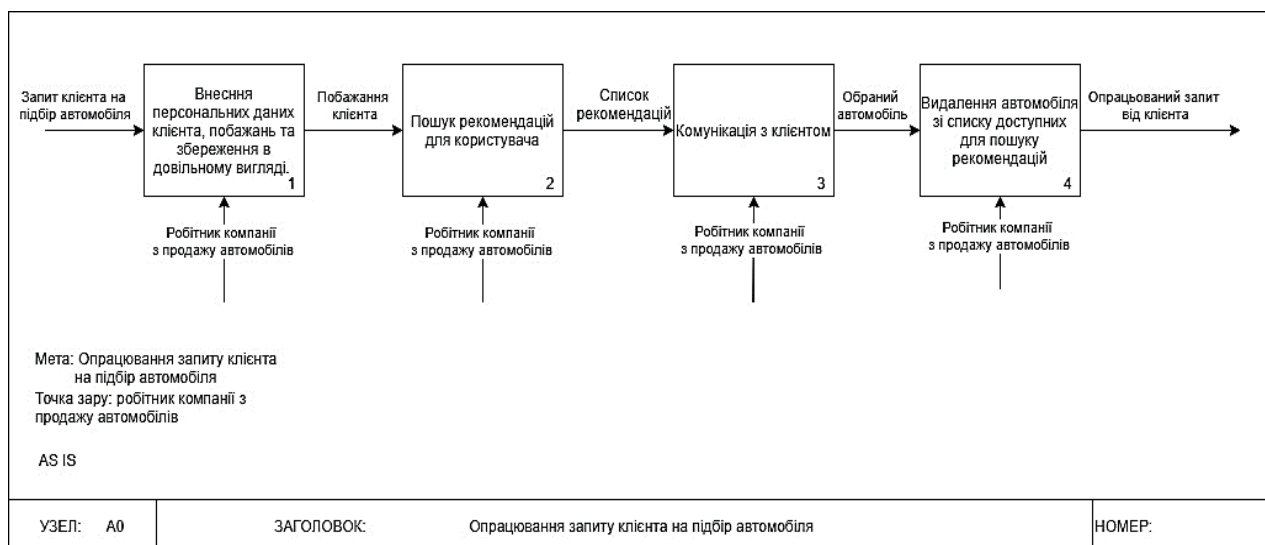


Рис. 1.1. Діаграма бізнес-процесу до впровадження рекомендаційної системи

Після впровадження системи рекомендацій змінюється бізнес-процес і більшість завдань виконуватиме система. За бажанням Замовника персональні дані та побажання Замовника зберігаються в системі. Після збереження даних співробітник компанії з продажу автомобілів починає співпрацю з конкретним

клієнтом, а після початку роботи система автоматично шукає рекомендації і змінює статус клієнта. Наступним кроком є спілкування з клієнтом для надання списку знайдених рекомендацій. Коли клієнт вибирає автомобіль, співробітник компанії з продажу автомобілів вибирає автомобіль в CRM-акаунті користувача. Якщо клієнт купує або відмовляється від покупки автомобіля, співробітник компанії змінює статус клієнта на «Купив автомобіль» або «Відмова від покупки». Якщо клієнт придбав автомобіль, його запис у CRM-системі деактивується та не враховується в подальших рекомендаціях. Якщо він відмовляється від покупки автомобіля, обраний дезактивований автомобіль буде активований. Бізнес-процес обробки запиту клієнта на вибір автомобіля після впровадження системи рекомендацій показано на рисунку 1.2 за допомогою діаграми TOBE IDEF0.

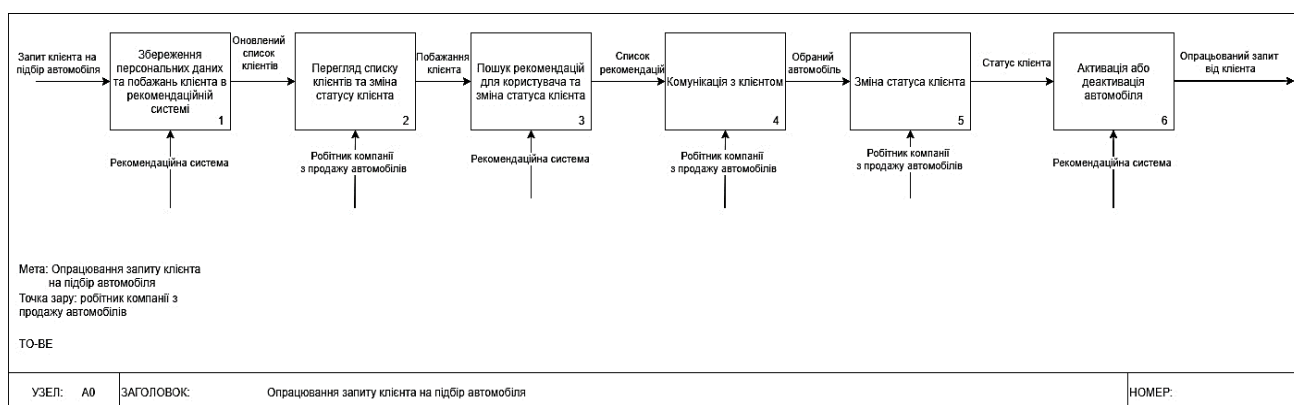


Рис. 1.2. Діаграма бізнес-процесу після впровадження рекомендаційної системи

1.2. Актори і функції

Перш ніж розробити діаграму варіантів використання, необхідно ідентифікувати акторів у системі, а потім визначити дії, які актори виконуватимуть. Система має двох акторів:

- клієнт;
- робітник компанії з продажу автомобілів.

Клієнт має можливість надати побажання щодо майбутнього автомобіля та особисті дані.

клієнта та даних про всі наявні в системі автомобілі (зі статусом «Активний»). Після того, як система знаходить рекомендації, статус клієнта змінюється на відповідний. Співробітник компанії з продажу автомобілів бачить, що статус клієнта змінився, і в формі для цього клієнта в таблиці відображаються автомобілі, які можуть йому підійти. Співробітник спілкується з клієнтом за телефоном або адресою електронної пошти, наданими клієнтом при створенні заявки. Після того, як клієнт зробить вибір, співробітник компанії з продажу автомобілів заповнює поле вибраного автомобіля, вибраний запис про автомобіль стає неактивним для пошуку рекомендацій для інших клієнтів, система змінює статус запису про клієнта на відповідний. Тоді співпраця з клієнтом закінчується. У випадку, якщо клієнт придбав автомобіль, співробітник компанії з продажу автомобілів змінює статус клієнта, автомобіль залишається недоступним для пошуку за рекомендаціями для інших клієнтів, створюється запис контактної сутності для зберігання даних клієнта, які можна використовувати надалі для маркетингових цілей. Якщо клієнт відмовляється від покупки автомобіля або автодилер не може з ним зв'язатися, співробітник змінює статус запису клієнта на відповідний, активується запис вибраного автомобіля, завдяки чому автомобіль буде використовуватися в пошуку рекомендацій для інших клієнтів.

1.4. Постановка задачі

Метою дослідження є підвищення ефективності роботи менеджерів, що в цілому збільшить прибуток компанії за рахунок збільшення загальної кількості оброблених заявок і продажів.

Мета системи – створити список рекомендованих автомобілів без урахування попередньої історії клієнта.

Для досягнення мети необхідно виконати наступні завдання:

- проаналізувати існуючі методи та інструменти та виявити проблеми в пошуку рекомендацій;
- створити та протестувати алгоритм пошуку рекомендацій;

- розробити архітектуру системи рекомендацій на платформі Dynamics 365;
- впровадження системи рекомендацій на платформі Dynamics 365.

1.5. Рішення з інформаційного забезпечення

Для зберігання даних використовується 4 сутності в Dynamics CRM:

- Contacts;
- Leads;
- Leads_Cars;
- Cars.

Сутність Contacts містить інформацію про персональні дані клієнтів після виконання роботи за заявкою клієнта, сутність Leads містить інформацію про персональні дані клієнта при подачі запиту та переваги вибору автомобіля, Leads_Cars є суб'єктом-посередником у з'єднанні Leads та Cars. Сутності Cars - це інформація про автомобілі. Сутності Contacts і Leads мають зв'язок «один до багатьох» або «нуль до багатьох».

У таблицях Lead і Cars існує зв'язок «багато до багатьох» і «один до нуля».

На рис. 1.4 зображено структурну схему бази даних.

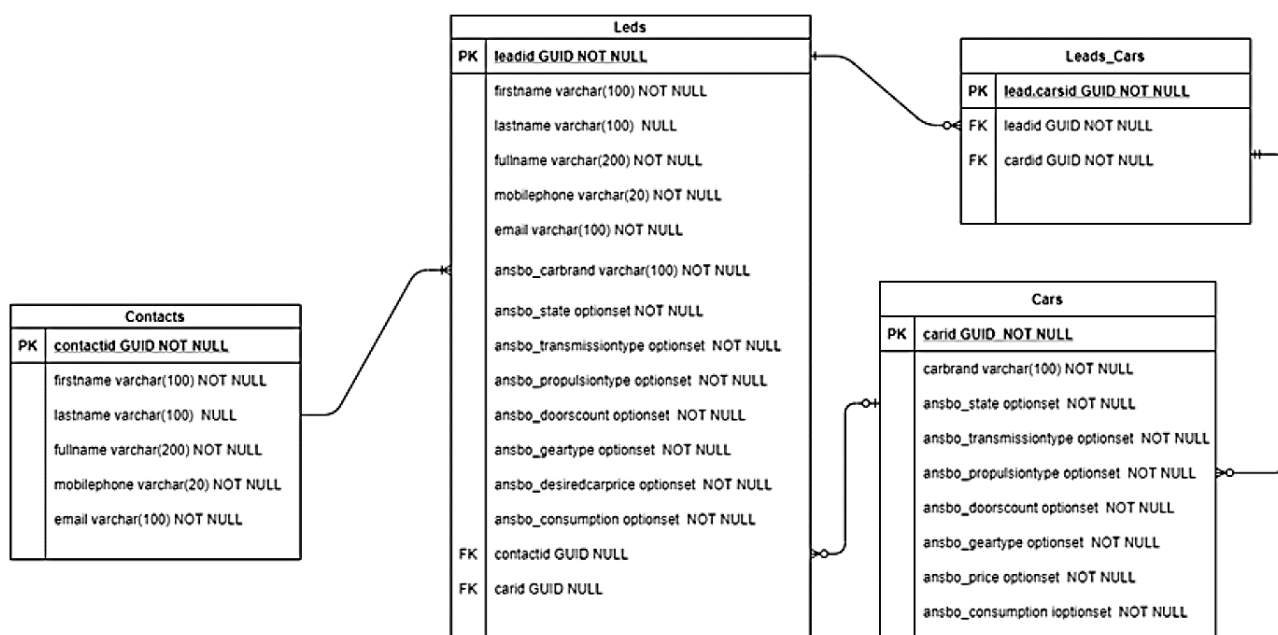


Рис. 1.4. Структурна схема бази даних

В табл. 1.1 описуються сутності та їх домени.

Таблиця 1.1

Опис структури використовуваних сутностей

Назва сутності	Опис сутності	Назва поля	Опис поля
1	2	3	4
Contacts	Сутність містить інформацію про персональні дані клієнтів після завершення роботи зі зверненням клієнта	contactid	Унікальний ідентифікатор клієнта
		firstname	Ім'я клієнта
		lastname	Прізвище клієнта
		fullname	Повне ім'я клієнта
		mobilephone	Мобільний телефон клієнта
		email	Емейл адреса клієнта
Leads	Сутність містить інформацію потенційних клієнтів	leadid	Унікальний ідентифікатор потенційного клієнта
		firstname	Ім'я клієнта
		lastname	Прізвище клієнта

1	2	3	4
		Fullname	Повне ім'я клієнта
		mobilephone	Мобільний телефон клієнта
Leads		email	Email адреса клієнта
		ansbo_carbrand	Бажаний бренд автомобіля
		ansbo_state	Стан автомобіля
		ansbo_transmissiontype	Тип коробки передач
		ansbo_propulsiontype	Бажаний тип рушійної сили
		ansbo_doorscount	Бажана кількість дверей
		ansbo_geartype	Бажаний тип приводу
		ansbo_desiredcarprice	Бажана ціна
		ansbo_consumption	Бажане споживання

1	2	3	4
		contactid	Посилання на запис сутності Контакт
		carid	Посилання на запис сутності Автомобіль
Leads_Cars	Проміжна сутність для зв'язку сутностей Lead та Car	lead.carsid	Унікальний ідентифікатор запису сутності
		leadid	Посилання на запис сутності Lead
		carid	Посилання на запис сутності Car
Cars	Сутність містить інформації про автомобілі	carid	Унікальний ідентифікатор автомобіля
		ansbo_carbrand	Бренд автомобіля
		ansbo_state	Стан автомобіля
		ansbo_transmissiontype	Тип коробки передач

Висновки до розділу 1

У частині, що описує конструктивні рішення для розробки системи рекомендацій підбору автомобілів для продажу клієнтам, визначено цілі дослідження, призначення системи та завдання, які необхідно виконати для досягнення мети. Також було описано бізнес-процес, розроблено блок-схему діяльності та діаграми IDEF0, що ілюструють структуру бізнес-процесів, розроблено структуру рішення інформаційного забезпечення у вигляді структурної діаграми бази даних, а також описано сутності та поля у формі столу.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Постановка задачі

Щоб здійснити вибір автомобіля, клієнт заходить на сайт компанії і заповнює характеристики автомобіля. Співробітник компанії приймає заявку і особисто шукає авто, які підійдуть клієнту. Оскільки замовник вибирає характеристики на свій розсуд, вони можуть конфліктувати для конкретного автомобіля. Співробітник компанії шукає рекомендації серед великої кількості автомобілів.

З наявних характеристик автомобілів, пропонованих компанією, необхідно знайти автомобілі з якомога більшою кількістю характеристик, які відповідають побажанням клієнта.

Дано:

1. Вектор характеристик, які обрав користувач $U = \{u_0, u_1, \dots, u_n\}$,

де u_j – певна характеристика, обрана користувачем;

n – загальна кількість характеристик,

$j=0, 1, \dots, n$;

2. Вектор характеристик для певного автомобіля $\Gamma_t = \{i_0, i_1, \dots, i_n\}$,

де i_j – певна характеристика автомобіля;

$j = 0, 1, \dots, n$;

n – загальна кількість характеристик;

$t = 1, 2, \dots, m$;

m – загальна кількість автомобілів.

Кількість бажаних характеристик користувача дорівнює кількості характеристик автомобілів.

Знайти:

Вектори ознак автомобілів з найменшим коефіцієнтом подібності.

2.2. Огляд методів розв'язання

Всесвітня павутина розвивається з кожним днем, збільшується кількість сайтів, сервісів і т. п. Окрім збільшення загальної кількості ресурсів, щоденно і мінімально збільшується кількість даних на кожному сайті. Якщо розглядати

такі послуги, як інтернет-магазини, то кількість товарів, які вони містять, постійно зростає і залежить від розробки нових продуктів у всіх сферах життя. Коли користувач заходить в інтернет-магазин і хоче вибрати певний товар, він витрачає багато часу на досягнення своєї мети. І час, необхідний для цього, з кожним днем збільшується, адже щодня асортимент магазину поповнюється тисячами новинок. Подібна ситуація і з іншими інтернет-ресурсами, поповненими терабайтами нової інформації. Системи рекомендацій допомагають користувачам мінімізувати час пошуку необхідної інформації, товарів і послуг.

Система рекомендацій -- це система, яка знаходить список елементів на основі певних критеріїв, які можуть бути релевантними для користувача. Дані, заповнені користувачем, дані про дії користувача, дані профілю користувача та іншу інформацію користувача можна використовувати для пошуку рекомендацій.

В даний час використання систем рекомендацій стає все більш популярним в ІТ-індустрії. Системи рекомендацій використовуються в електронній комерції для пошуку продуктів, які можуть зацікавити користувачів, у потокових системах для рекомендації цікавих трансляцій, фільмів, каналів і в маркетингу для пошуку цільових груп користувачів. Крім того, браузері використовують системи рекомендацій, вбудовані в пошукові системи, щоб знаходити точніші результати за запитом користувача, фільтруючи заборонений вміст і спам. Поштові клієнти також використовують системи рекомендацій для класифікації вхідних повідомлень у групи.

Рекомендації діляться на два види:

- персоналізовані рекомендації;
- неперсоналізовані рекомендації.

Існують підходи до пошуку персоналізованих рекомендацій, такі як фільтрація вмісту даних, спільна фільтрація та гібридна фільтрація, які поєднують попередні два підходи.

Основна ідея контент-орієнтованих підходів полягає в тому, щоб знайти подібні характеристики предметів, вибраних конкретним користувачем, щоб рекомендувати йому товари, схожі на ті, які користувач раніше оцінив добре. Недоліком підходів на основі контенту є те, що вони погано працюють у пошуку рекомендацій, якщо система рекомендацій, яка використовує цей підхід, має обмежену кількість даних і невелику кількість характеристик користувачів і елементів [5].

На відміну від підходів, заснованих на вмісті, методи спільної фільтрації знаходять рекомендації на основі оцінок елементів, наданих іншими користувачами в системі. Цей підхід ґрунтується на припущенні, що два користувачі повинні мати однакові оцінки для елемента, якщо обидва користувачі дали високу оцінку тому самому елементу. Крім того, підхід спільної фільтрації дозволяє диверсифікувати рекомендації завдяки тому факту, що кожен користувач має власні вподобання, і схожі користувачі можуть мати одні схожі рекомендації, а інші зовсім не такі. Однак завдяки таким уподобанням користувач може отримати рекомендацію щодо теми, яка його зовсім не цікавила, але може зацікавити [5].

Методи спільної фільтрації поділяються на два види:

- методи, засновані на пошуку сусідів;
- методи, засновані на модельному навчанні.

Методи сусідства використовують наявні в системі рейтинги користувачів і продуктів. Ці методи поділяються на два види:

- орієнтовані на користувача;
- об'єктно-орієнтовані.

Методи пошуку сусідів, орієнтовані на користувача, знаходять продукти, які можуть підійти конкретному користувачеві, на основі оцінок цих продуктів іншими користувачами.

При використанні орієнтованих на продукт методів виявлення сусідів суб'єктом рекомендації є сам користувач. Користувачі, які дали хороші оцінки

продукту, подібному до продукту, для якого шукаються рекомендації, з'являються в нових функціях певного продукту. Характерною рисою цих методів є цільовий об'єкт для пошуку рекомендацій, а спільною ознакою є те, що вони шукають рекомендації за оцінками користувачів, наявними в системі [5].

Методи, засновані на навчанні моделі, використовують наявні дані в системі для навчання моделі. Потім даються рекомендації щодо навченої моделі. Однак цей підхід вимагає достатньо великого набору вихідних даних, щоб процес пошуку рекомендацій був точним. Вони також використовують практику перенавчання моделі з часом, щоб підвищити точність пошуку рекомендацій. Такий підхід можливий завдяки тому, що база даних системи постійно поповнюється новими оцінками користувачів [5].

2.3. Рекомендаційні системи

Рекомендаційні системи дуже актуальні в наш час, оскільки все більше послуг і галузей починають їх використовувати, тому з'явилося багато наукових робіт на цю тему.

У статті [5] розглядаються два методи створення рекомендаційних мереж: фільтрація вмісту та спільна фільтрація. Обговорювані методи базуються на аналізі профілів користувачів або поведінки користувачів. Загальний метод створення мережі рекомендацій, основні кроки підходу на основі сусідства та підходу спільної фільтрації на основі моделі обговорюються більш детально. Також проведено порівняння розглянутих методів створення рекомендаційної мережі.

Стаття [6] обговорює різні підходи до створення мережі рекомендацій. Розглядаються підходи до збору даних користувачів, фільтрації вмісту та спільної фільтрації. У статті також розглядаються та описуються основні проблеми систем рекомендацій: проблема холодного старту, бульбашки фільтра. При розгляді фільтрації вмісту враховуються такі методи класифікації, як класифікатори на базі мережі Бейса, класифікатори на основі нейронної

мережі, класифікатори на основі дерева рішень і класифікатори на основі алгоритму кластеризації. Також обговорюються підходи до спільної фільтрації. При розгляді кооперативної фільтрації були враховані методи розрахунку коефіцієнтів подібності, такі як Евклідова відстань, Манхеттенська відстань і коефіцієнт кореляції Пірсона.

У статті [7] розглядаються системи рекомендацій, що використовуються в соціальних мережах. Побудовані моделі у вигляді графа використовуються для пошуку рекомендацій. Для пошуку публікацій, рекомендованих конкретним користувачам у соціальній мережі, використовується стохастичний граф, вершинами якого є користувачі та публікації, а ребрами – дії користувача над конкретними публікаціями. Для пошуку рекомендацій побудований графік аналізується.

У статті [8] розглядаються основні методи пошуку рекомендацій та більш детально описується гібридний метод пошуку рекомендацій. Включає спільну фільтрацію та фільтрацію на основі вмісту. У статті також розглянуто основні види гібридних систем рекомендацій:

- система рекомендацій за методом спільної фільтрації з додатковими можливостями методу контентної фільтрації;
- система рекомендацій за методом фільтрації з додатковими можливостями методу спільної фільтрації.

Стаття [9] стосується проблеми неможливості надання рекомендацій для користувачів, для яких немає історії використання системи (немає профілів). У статті запропоновано використання гібридного підходу, який враховує не лише вектори контентних особливостей об'єктів, а й вектори демографічних особливостей користувачів.

У статті [10] детально розглядається основна проблема систем рекомендацій – проблема холодного старту. У статті вказано основні причини даної проблеми, вказано на можливу циклічність даної проблеми, а також наведено основні методи її вирішення:

- гібридизація системи рекомендацій за допомогою спільної фільтрації та фільтрації на основі контенту;
- використання характеристик користувача при пошуку рекомендацій (регіон, країна, дата та час, часовий пояс тощо).

У статті [11] представлено класифікацію систем рекомендацій, яка відрізняється від класифікацій, розглянутих у попередніх статтях. Автори поділяють системи рекомендацій на дві категорії:

- системи рекомендацій, що використовують алгоритми на основі пам'яті;
- системи рекомендацій з використанням модельних алгоритмів.

Алгоритми на основі пам'яті використовують усі існуючі рейтингові дані та шукають рекомендації, зіставляючи конкретного користувача з наявними даними інших користувачів.

Алгоритми на основі моделі використовують параметризовану модель. Вони використовують дані на основі оцінок інших користувачів для налаштування моделі. Процес пошуку рекомендацій керується моделлю.

Найпопулярніші алгоритми на основі пам'яті включають:

- загальні алгоритми фільтрації на основі даних користувача;
- загальні алгоритми фільтрації на основі даних про товар.

Ці алгоритми знаходять рекомендації на основі рейтингів найбільш схожих користувачів або продуктів. Алгоритми спільної фільтрації ґрунтуються на припущенні, що якщо схожі користувачі оцінюють один товар однаково на основі певної кількості параметрів, ті самі користувачі повинні оцінювати інші елементи подібним чином. Це твердження працює аналогічно. Якщо одному користувачеві подобаються два схожих предмета за певною кількістю параметрів, то іншому вони теж повинні сподобатися.

У випадку алгоритмів на основі моделі основним принципом є попереднє створення та навчання моделі. Коли модель навчена на тестових даних, її можна використовувати для пошуку рекомендацій для інших користувачів.

Тестові дані – це дані оцінок продукту від користувачів.

У статті [12] розглядається пошук подібності між об'єктами шляхом пошуку подібності між їхніми описами. Опис кожного елемента перетворюється на вектор слів, а для кожного опису елемента обчислюється зворотний частотний вектор (TF-IDF). Частота – це відносна частота певного терміну. Інверсна частота документа – це відносна кількість векторів опису, які містять слово як елемент. Для побудови матриці використовується добуток відносної частоти слів і зворотної частоти документа. Побудова цієї матриці необхідна для того, щоб слова, які найчастіше використовуються в описах об'єктів, мали менший вплив на пошук рекомендацій, ніж ті, які зустрічаються рідше, і тому допомагали точніше знаходити найбільш схожі об'єкти. На основі отриманої матриці автори використовують косинусну подібність для знаходження найбільш схожих об'єктів.

У статті [13] розглядається побудова ситуативної моделі вибору користувача. Рекомендації складаються на основі підходу, орієнтованого на схожість користувачів. Модель являє собою дводольний багат шаровий граф. Кожен рівень представляє переваги користувача в певний момент часу. Вершинами графа є користувачі та об'єкти. Зв'язок між шарами відбувається як між вершинами об'єктів. Кожен шар містить зв'язки між вершинами користувача та вершинами елементів. Дуги, що з'єднують вершини, залежать від явного або неявного зворотного зв'язку. Якщо зв'язок визначається на основі оцінки користувача, то це очевидно. Якщо зв'язок заснований на виборі користувача, він прихований. Важливим аспектом запропонованого підходу є те, що багат шаровий характер графіка дозволяє відстежувати зміни у виборі клієнтів і рейтингах з часом, а також дозволяє аналізувати тенденції у виборі, зміни в уподобаннях користувачів і надавати точніші рекомендації.

Стаття [14] обговорює підходи до отримання вихідних даних для пошуку рекомендацій. Основні підходи такі:

- чіткий зворотний зв'язок;
- прихований зворотний зв'язок;
- гібридний зворотний зв'язок.

Під час використання явного зворотного зв'язку система пропонує користувачеві через системний інтерфейс надати оцінки елементів, щоб побудувати та вдосконалити свою модель. Точність рекомендацій залежить від кількості оцінок, наданих користувачем. Недоліком цього методу є те, що він вимагає зусиль з боку користувачів, і користувачі не завжди готові надати достатньо інформації. Незважаючи на те, що однозначний зворотній зв'язок вимагає від користувача більше зусиль, він вважається джерелом найбільш надійних даних, оскільки не передбачає його використання, а також забезпечує прозорість процесу рекомендацій, що призводить до більшого прийняття рекомендацій і збільшення за рівнем довіри до рекомендацій.

При використанні явного зворотного зв'язку система автоматично визначає вподобання користувача. Уподобання визначаються шляхом відстеження активності користувача: відстеження історії покупок, історії перегляду, аналізу потоку кліків, аналізу запитів, аналізу вмісту електронної пошти, аналізу натискання кнопок тощо. Прихований зворотний зв'язок не вимагає ручної оцінки продуктів, але аналізує переваги на основі історії дій у системі. Метод менш точний, ніж явний зворотний зв'язок, оскільки він може враховувати дії, які користувач виконав випадково, але в той же час такі переваги є більш об'єктивними, оскільки переваги виникають автоматично на основі дій, і користувач не має можливості упереджено знизити або підвищити рейтинг тих чи інших товарів через власні переконання.

Гібридний зворотний зв'язок включає явну та неявну інформацію. Цей метод є найкращим варіантом, оскільки ручне оцінювання продуктів передбачає створення параметрів на основі вашої історії активності. Цей тип зворотного зв'язку можна налаштувати залежно від потреб ваших систем. Наприклад, ручне оцінювання певного продукту може бути необов'язковим, але застосовуватись лише тоді, коли користувач вирішить оцінити його.

У статті також пропонується періодичне навчання системи на основі отриманих даних зворотного зв'язку, що дозволить підвищити рівень пошуку рекомендацій.

У статті [15] розглядаються системи рекомендацій у сфері індивідуальних цін. Ці типи систем рекомендацій включають відображення знижок на продукти лише для певних груп клієнтів. Клієнти поділяються на наступні групи:

- байдужі;
- чутливі до знижок;
- стандартні.

Покупці, що належать до байдужої групи, не купують товари зі знижкою чи без неї.

Клієнти, чутливі до знижок, зазвичай купують лише продукти зі знижкою.

Покупці стандартної групи готові купувати товари за стандартною ціною, без знижок.

Метою системи рекомендацій є знаходження групи користувачів, рекомендованих замовнику за його характеристиками. Завдяки такому розподілу власники системи рекомендацій отримують максимальний прибуток, оскільки знижки отримують лише чутливі до знижок клієнти, а всі інші купують товар за стандартною ціною.

У статті [16] розглядаються характеристики надійності рекомендаційних систем та методи ідентифікації атак на рекомендаційні системи.

Основним показником, який використовується для оцінки надійності системи рекомендацій, є помилка передбачення. Помилка передбачення – це міра того, наскільки змінилися рекомендації до та після атаки. Однак може статися так, що рейтинг деяких продуктів буде дуже низьким, і тому даний продукт, рейтинг якого змінився під час атаки, все одно не буде включено до рекомендацій. Для підвищення точності оцінки надійності використовується додатковий показник – коефіцієнт виклику. Для оцінки надійності системи створено дві вибірки: вибірка з реальними користувачами та вибірка з користувачами, що імітують атаку. Наведені показники розраховуються, а результати аналізуються та порівнюються.

Для виявлення атаки на систему рекомендацій здійснюється пошук і видалення з системи профілів ботів, а також усіх залишених у системі даних: рекомендацій, коментарів, оцінок. Для пошуку профілів ботів використовується класифікатор на основі алгоритмів машинного навчання.

Стаття [17] обговорює побудову правил асоціації для пошуку рекомендацій. Типова матриця користувачів і предметів, які вони оцінюють, замінюється матрицею користувачів і їхніх інтересів. Для користувачів, які вже мають історію рейтингів за допомогою правил асоціації, є додаткові рекомендації. Для нового користувача доступний список із, наприклад, десяти елементів із найвищими оцінками інших користувачів. На основі першого вибору користувача створюються правила асоціації для подальшого пошуку рекомендацій.

Стаття [18] обговорює пошук рекомендацій для спільноти користувачів. Щоб знайти рекомендації для групи користувачів, весь набір користувачів, що міститься в даній системі, необхідно розділити на групи схожих користувачів. Мірою подібності між користувачами є евклідова відстань між векторами ознак користувача. Автори розглядають використання методу k -середніх для кластеризації кількох користувачів. Недоліком методу k -середніх є те, що він локально збіжний і вимагає визначення початкової кількості кластерів і центрів кластерів. Після поділу набору на кластери обчислюється сумарний рейтинг предметів цієї групи користувачів. Коли вони знаходять нового клієнта, вони шукають групу користувачів, на яких вони найбільше схожі, і дають рекомендації на основі їхнього сукупного рейтингу.

У статті [19] обговорюється модифікація методу спільної фільтрації шляхом введення релевантності. З'явився відносно класичний алгоритм оцінки релевантності продукту з модифікацією. Класичний алгоритм враховує показник людей, які купили або не купили товар, тих, хто відвідав або не відвідав сторінку товару тощо. Автори пропонують використовувати замість двійкового показника час, який клієнт витратив на товар. сторінку слід використовувати як міру інтересу користувача до конкретного продукту. Якщо

час перевищує певне порогове значення, то продукт релевантний, інакше він нерелевантний. Використання цього індикатора забезпечує точніший пошук рекомендацій, оскільки замість щойно придбаних товарів система запропонує товари, схожі на ті, які користувач переглядав найдовше.

У статті [20] розглядається система рекомендацій, яка враховує емоційний тон коментарів користувачів. При побудові систем рекомендацій часто використовуються тільки позитивні відгуки. Користувачі залишають «лайк» або взагалі нічого не залишають. Такий підхід втрачає відгук від багатьох користувачів, тому що користувач не хоче ставити «лайк» і взагалі нічого не залишає, але може написати коментар, тому аналіз емоційної тональності коментарів є досить непоганим показником у поєднанні з оцінками, або за відсутності оцінок. Щоб заповнити порожні оцінки для деяких елементів, емоційний тон відгуку чи коментаря буде перетворено на оцінку за допомогою машини факторизації, а контекстна інформація відповіді буде використана як додатковий параметр машини факторизації.

У статті [21] розглядаються системи рекомендацій, побудовані на основі онтологій. Онтологія – це загальні поняття тематичних областей. Онтологія використовується для формалізації певної тематичної області шляхом поділу її на певний перелік об'єктів і зв'язків між ними. Щоб знайти рекомендації, система рекомендацій виконує спільний запит до доступних онтологій і намагається знайти елементи, які задовольняють усім обмеженням, вибраним користувачем. Якщо жодного запису не знайдено, система запитує список онтологій із спадним числом обмежень, прийнятих із наданого клієнтом списку обмежень. Система поступово зменшує кількість обмежень, поки не знайде хоча б один елемент, який відповідає обмеженням. Коли результат знайдено, якщо не знайдено жодної рекомендації для всіх обмежень, система повідомляє користувача, що деякі обмеження не були включені в пошук, і запитує, чи відобразити результат. У разі згоди замовника відображається список рекомендацій, а також список обмежень, виключених у пошуку рекомендацій. Користувач може змінити ці обмеження та повторити пошук.

Стаття [22] стосується системи рекомендацій, яка містить два рівні пошуку рекомендацій. Система рекомендацій повинна знаходити елементи на першому рівні, а потім на основі знайдених рекомендацій рекомендувати розділити користувачів на групи, яким можуть бути цікаві предмети, знайдені на першому рівні. Для вирішення проблеми знаходження обмежень для поділу користувачів на групи пропонується підхід, у якому обмеження для поділу користувачів визначаються в термінах нечітких наборів, а потім застосовується фазоутворення для перетворення нечітких наборів обмежень у чіткі параметри для поділу користувачів на групи.

У статті [23] обговорюється, як зберігати величезні обсяги даних у системах рекомендацій. Оскільки алгоритми видобутку рекомендацій зазвичай використовують і обробляють великі обсяги даних, необхідно мінімізувати час виконання запитів на зберігання даних, оскільки час роботи алгоритмів безпосередньо залежить від часу обробки інформації. Автори пропонують використовувати бази даних NoSQL. Переваги баз даних NoSQL включають, перш за все, те, що вони характеризуються значно вищою ефективністю, близькою до лінійної масштабованістю та відсутністю прозорої схеми даних. Перевагою таких баз даних також є вбудований паралелізм запитів.

У статті [24] розглядається побудова системи рекомендацій з даними користувача, які не є частиною бази даних системи рекомендацій. Для аналізу вподобань автори пропонують використовувати профілі користувачів у соціальних мережах. Такий підхід дозволить зменшити обсяг системної бази даних і вирішити проблему наповнення даними профілю користувача. Однак такий підхід вимагає постійної доступності серверів соціальних мереж, а також попереднього збору даних користувачів перед пошуком рекомендацій.

Висновки до розділу 2

У результаті аналізу виявилось, що не врахована ситуація, коли для пошуку рекомендацій необхідно викласти рекомендації серед існуючих об'єктів на основі побажань користувача. Ця ситуація важлива, тому що часто буває так, що користувач не має профілю на даному ресурсі, вперше заходить

туди, де надаються конкретні послуги, але хоче отримати рекомендацію, наприклад, купити певний товар, вибрати туристичний пакет, вибір фільму в кінотеатрі, користування послугою тощо. Тому важливою є обрана сфера дослідження та створення системи рекомендацій.

РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Розробка алгоритму пошуку рекомендацій

Для побудови системи рекомендацій був використаний підхід до фільтрації вмісту даних [5] відповідно до особистих побажань замовника. Наявні в системі автомобілі вважаються елементами алгоритму рекомендаційного пошуку, користувачами – запити клієнтів, що містять бажані характеристики автомобіля. Розглянемо блок-схему алгоритму пошуку рекомендацій на рис. 3.1.

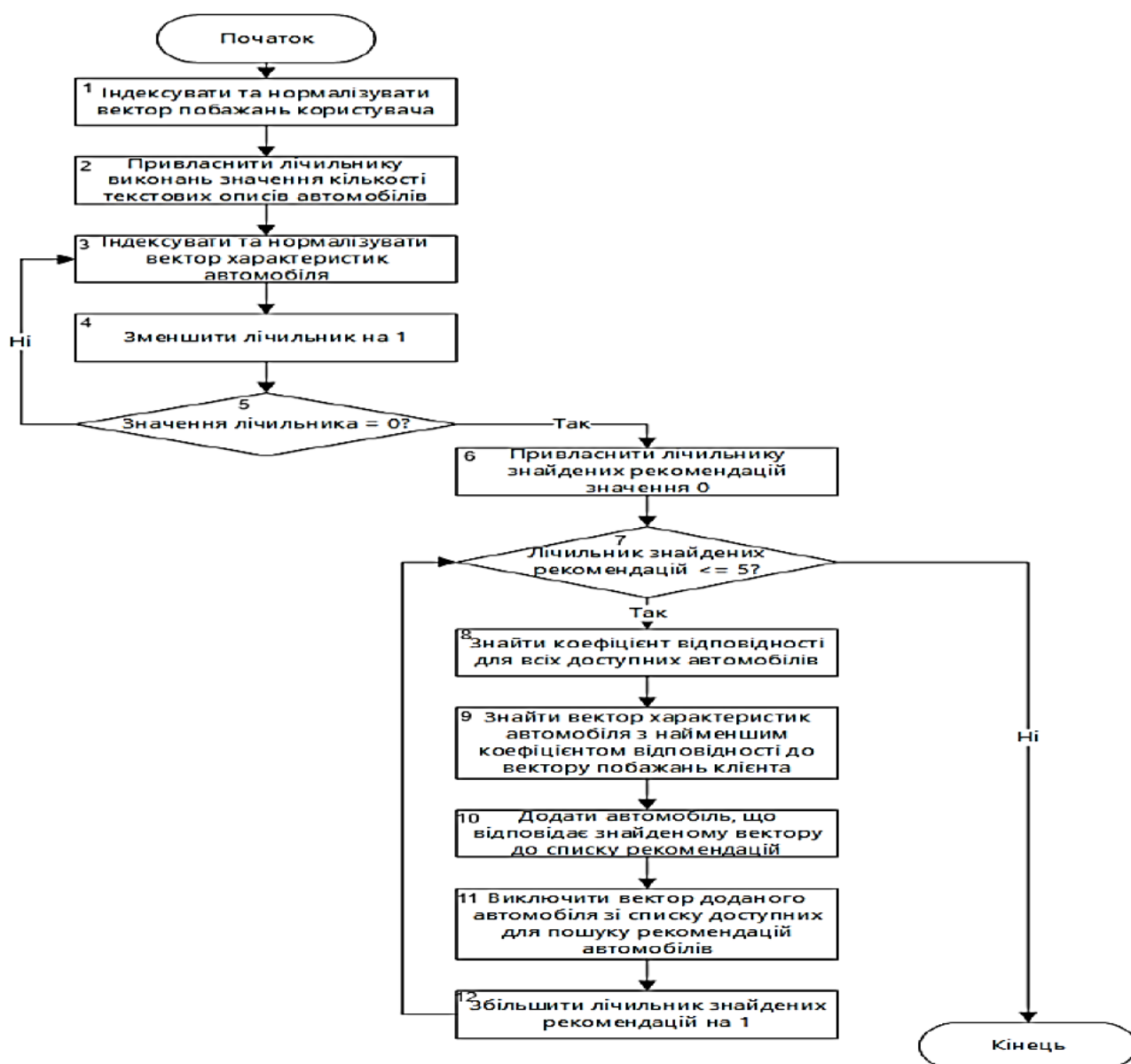


Рис. 3.1. Блок-схема алгоритму пошуку рекомендацій

Першим етапом роботи алгоритм передбачає перетворення текстових характеристик бажаних параметрів автомобіля, отриманих із програми клієнта та характеристик автомобіля, у вектори слів, індексацію та нормалізацію отриманих векторів. Перший етап алгоритму показано на кроках 1-5 на блок-схемі алгоритму, представлений на рис. 3.1 [4].

Другим етапом роботи алгоритм передбачає пошук рекомендацій. Процес пошуку рекомендацій триває, доки не буде знайдено певну кількість автомобілів, які відповідають запиту. Другий етап алгоритму показаний на кроках 6-12 на блок-схемі алгоритму, зображеній на рис. 3.1 [4].

Щоб знайти список предметів (автомобілів), які підходять для користувача, відфільтруємо предмети, описані векторами атрибутів, щоб вони задовольняли переваги конкретного користувача. У цьому випадку основна ідея алгоритму полягає в пошуку подібності між об'єктом користувача та елементами, які мають різну семантику, але знаходяться в одному евклідовому n -вимірному просторі. Кореляція між векторами атрибутів об'єкта користувача та елемента представлена як відстань. Оцінка подібності, представлена евклідовою відстанню між атрибутами запиту користувача та атрибутами елемента, обчислюється на основі найбільш подібних атрибутів (наприклад, які мають найбільшу відстань між ними) [6]. Це означає, що подібність векторів побажань користувача та векторів характеристик автомобіля виходить шляхом обчислення суми квадратів часткових відстаней між окремими атрибутами.

Алгоритм знаходить елементи, які є найбільш релевантними для конкретного елемента користувача відповідно до вектора атрибутів \bar{U} (вектор переваг користувача) і створює кластер, де конкретний користувач пов'язаний з низкою елементів. Подібність користувачів та елементів представлена Евклідовою відстанню між векторами атрибутів користувача U та атрибутів елемента Γ . Кількість атрибутів вектору користувача та векторів автомобілів є однаковою.

Подібність знаходимо за формулою:

$$S_w^2(\bar{U}, \bar{I}_t) = \sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2 \right), \quad (3.1)$$

де $S_w^2(U, I_t)$ – зважений коефіцієнт подібності між користувачем та елементом,

$i_n^{(t)}$ – значення n -го атрибута t -го елемента,

U_n – значення n -го атрибута для конкретного користувача,

r_n – коефіцієнт подібності між n -м атрибутом з вектору атрибутів користувача та вектором елементів,

N_p – загальна кількість атрибутів користувачів та елементів.

На відміну від звичайної формули евклідової відстані, яка дозволяє обчислити фактичну відстань між двома векторами атрибутів, коефіцієнт подібності отримується як зважене значення відстані, обчислене як сума квадратів часткових відстаней між конкретними атрибутами користувача або елемента, помножених на за деяким ваговим коефіцієнтом r . Тобто знаходиться подібність, яка базується на відстані та значення якої залежить від кількох параметрів.

У цьому випадку, крім часткової відстані між атрибутами користувача та елемента, використовується параметр лексикографічної подібності між цими атрибутами r_n . Оскільки реалізується контекстна модель даних, що буде використовувати для пошуку рекомендацій використовуємо значення лексикографічної подібності r_n , як параметр формули (3.1)

$$r_n = \sum_j \left(\frac{1}{\min\{L\}} | i_j^{(n)} - u_j^{(n)} | \right), \quad (3.2)$$

де r_n – коефіцієнт лексикографічної подібності між n -м атрибутом конкретного користувача та елемента,

$i_j^{(n)}$ – j -ий символ у рядковому поданні n -го атрибута конкретного елемента,

$u_j^{(n)}$ – j -ий символ у рядковому поданні n -го атрибута конкретного користувача,

$\min\{L\}$ – фактична довжина найменшого з двох рядків, що представляють n -й атрибут користувача або елемента.

Щоб обчислити подібність атрибутів користувача або елемента, будемо використовувати лінійний алгоритм пошуку, повторюючи два рядки, що представляють n -й атрибут, і для кожної позиції порівнюємо символи, розташовані на тій самій j -й позиції. Якщо ці символи $i_j^{(n)}$ і $u_j^{(n)}$ точно збігаються, то додаємо значення L до значення подібності r_n . Параметр L – це фактична довжина найменшого рядка, яка представляє n -й атрибут користувача або елемента.

Крім того, щоб забезпечити більш гнучкі обчислення подібності відстані за допомогою формул (3.1, 3.2), формула (3.1) змінена з використанням іншого додаткового параметра m , що представляє фактичну кількість атрибутів користувача або елемента, що відповідають значенню коефіцієнта подібності $r_n=1$. Основною причиною використання додаткового параметра є те, що схожість конкретних користувачів та елементів, представлена значенням фактичної відстані між вектором атрибутів користувача та елемента, багато в чому залежить від кількості атрибутів, які є лексикографічно рівними. Тоді формула (3.1) матиме вигляд:

$$S_w(\bar{U}, \bar{I}_t) = \frac{1}{m} * \sqrt{\sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2 \right)}. \quad (3.3)$$

Велика кількість атрибутів з лексикографічною подібністю робить евклідову відстань між двома векторами атрибутів користувача або елемента малою, таким чином збільшуючи подібність між конкретним користувачем і елементом. Однак бувають випадки, коли конкретний користувач і елемент не мають атрибутів, схожих на значення додаткового параметра $m = 0$. У цьому випадку будемо приймати значення параметра $m = 0.01$, що забезпечує велику відстань між радикально іншим користувачем і елементом, щоб уникнути випадків, коли елементи з атрибутами, які абсолютно несумісні з атрибутами користувача, мають ближчу відстань до цього користувача [3]. На блок-схемі рис. 3.1 представлена схема роботи алгоритму.

3.2. Приклад роботи алгоритму

Алгоритм приймає на вхід текстовий опис побажань користувача та текстовий опис характеристик автомобіля. Приклад тестового опису побажань користувача, отриманих від системи, наведено в табл. 3.1, а зразковий опис тестування характеристик автомобіля, отриманих із системи, наведено в табл. 3.2.

Таблиця 3.1

Текстовий опис побажань користувача, отриманих з системи

Користувач	bmw V8 less_than_thirty Hybrid two full less_than_fifteen
------------	---

Таблиця 3.2

Текстовий опис характеристик автомобілів, отриманих з системи

Audi A8	audi V6 less_than_thirty Fuel four back less_than_fifteen
BMW I7	bmw V8 less_than_forty Hybrid two back less_than_fifteen

Для кожного автомобіля текстові описи ознак розбиваються на вектори ознак, слова індексуються, а отримані показники нормалізуються.

У табл. 3.3 наведені знайдені значення атрибутів вектору характеристики автомобіля Audi A8.

Таблиця 3.3

Індекси атрибутів вектору характеристик для автомобіля Audi A8

audi	1
V6	2
less_than_thirty	3
Fuel	4
four	5
back	6
less_than_fifteen	7

У табл. 3.4 наведено індекси атрибутів вектору характеристик для автомобіля Audi A8 після нормалізації

Таблиця 3.4

Нормалізовані індекси атрибутів вектору характеристик для автомобіля Audi A8

audi	1
V6	0.076985645933014352
less_than_thirty	0.14875598086124403
Fuel	0.14875598086124403
four	0.29229665071770339
back	0.36406698564593304
less_than_fifteen	0.43583732057416269

У табл. 3.5 наведено знайдені індекси атрибутів вектору характеристик для автомобіля BMW I7.

Таблиця 3.5

Індекси атрибутів вектору характеристик для автомобіля BMW I7

bmw	8
V8	9
less_than_forty	10
Hybrid	11
two	12
back	6
less_than_fifteen	7

У табл. 3.6 наведено індекси атрибутів вектору характеристик для автомобіля Audi A8 після нормалізації

Таблиця 3.6

Нормалізовані індекси атрибутів вектору характеристик для автомобіля BMW I7

bmw	0.50760765550239229
V8	0.579377990430622
less_than_forty	0.6511483253588517
Hybrid	0.7229186602870814
two	0.794688995215311
back	0.36406698564593304
less_than_fifteen	0.43583732057416269

Для кожного елемента з вектору атрибутів користувача перевіряється, чи з'являється той самий атрибут у векторі транспортного засобу. Якщо той самий атрибут знайдено, йому буде присвоєно значення індексу існуючого елемента. Якщо такого значення немає, індекс атрибута у векторі характеристики вставляється у вектори характеристики автомобіля, а потім показники нормалізуються. У табл. 3.7 наведено знайдені атрибутивні показники вектору бажань користувача.

Таблиця 3.7

Індекси атрибутів вектору побажань користувача

bmw	0.50760765550239229
V8	0.579377990430622
less_than_thirty	0.14875598086124403
Hybrid	0.7229186602870814
two	0.794688995215311
full	6
less_than_fifteen	0.43583732057416269

У табл. 3.8 наведено знайдені індекси атрибутів вектору побажань користувача після процесу нормалізації.

Нормалізовані індекси атрибутів вектору побажань користувача

bmw	0.50760765550239229
V8	0.579377990430622
less_than_thirty	0.14875598086124403
Hybrid	0.7229186602870814
two	0.794688995215311
full	0.69899521531100473
less_than_fifteen	0.43583732057416269

Після підготовки даних для вектору атрибутів бажань користувача визначається зважений коефіцієнт подібності для кожного вектору атрибутів характеристики автомобіля. Знаходиться автомобіль із мінімальним значенням зваженого коефіцієнта подібності.

Цей процес повторюється до тих пір, поки не буде знайдено кількість автомобілів, що відповідає необхідній кількості рекомендацій. У нашому випадку нам потрібно знайти одну рекомендацію, щоб рекомендований автомобіль був автомобілем з меншим зваженим коефіцієнтом подібності.

В табл. 3.9 наведено розрахунок коефіцієнта лексикографічної подібності, добуток коефіцієнта лексикографічної подібності на квадрат різниці між значеннями відповідних значень атрибута бажання користувача і вектору ознак автомобіля Audi A8, загальне кумулятивне значення для всього вектору автомобіля.

Таблиця 3.9

Розрахунок значень для знаходження зваженого коефіцієнту подібності між вектором користувача та вектором автомобіля Audi A8 (програмне обчислення)

Атрибут побажань користувача	Атрибут автомобіля	$\frac{1}{r_n}$	$\frac{1}{r_n} * [i_n^{(t)} - U_n]^2$	$\sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2 \right)$
bmw	audi	0	0.2424502209198508	0.2424502209198508
V8	V6	0.5	0.12619903390490142	0.36864925482475219
less_than_thirty	less_than_thirty	1	0	0.36864925482475219
Hybrid	Fuel	0	0.25239806780980295	0.62104732263455514
two	four	0	0.25239806780980295	0.873445390444358
full	back	0	0.112176919026579	0.985622309470937
less_than_fifteen	less_than_fifteen	1	0	0.985622309470937

Знаходимо коефіцієнт подібності для вектору характеристик автомобіля Audi A8:

$$S_w(\bar{U}, \bar{I}_t) = \frac{1}{m} * \sqrt{\sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2 \right)} = 0.49639256377159224.$$

В табл. 3.10 наведено розрахунок коефіцієнта лексикографічної подібності, добутку коефіцієнта лексикографічної подібності на квадрат різниці між значеннями відповідних значень атрибута бажання користувача і вектору ознак автомобіля BMW I7, сума кумулятивного значення для всього вектору автомобіля.

Розрахунок значень для знаходження зваженого коефіцієнту подібності між вектором користувача та вектором автомобіля BMW I7 (програмне обчислення)

Атрибут побажань користувач a	Атрибут автомобіля	$\frac{1}{r_n}$	$\frac{1}{r_n} * [i_n^{(t)} - U_n]^2$	$\sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2 \right)$
bmw	bmw	1	0	0
V8	V8	1	0	0
less_than_th irty	less_than_f orty	0.66666666 666666663	0.168265378539 86862	0.168265378539868 62
Hybrid	Hybrid	1	0	0.168265378539868 62
two	two	1	0	0.168265378539868 62
full	back	0	0.112176919026 579	0.280442297566447 62
less_than_fi fteen	less_than_fi fteen	1	0	0.280442297566447 62

Знаходимо коефіцієнт подібності для вектору характеристик автомобіля BMW I7:

$$S_w(\bar{U}, \bar{I}_t) = \frac{1}{m} * \sqrt{\sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2 \right)} = 0.10591360584296006$$

Рекомендованим користувачу буде автомобіль BMW I7 тому, що він має менше значення коефіцієнта подібності.

3.3. Результати експериментальних досліджень

Час роботи алгоритму досліджувався по мірі збільшення кількості автомобілів, серед яких необхідно було знайти рекомендації.

На рис. 3.2 наведено графік часу роботи алгоритму в залежності від кількості автомобілів, серед яких знайдено рекомендації.

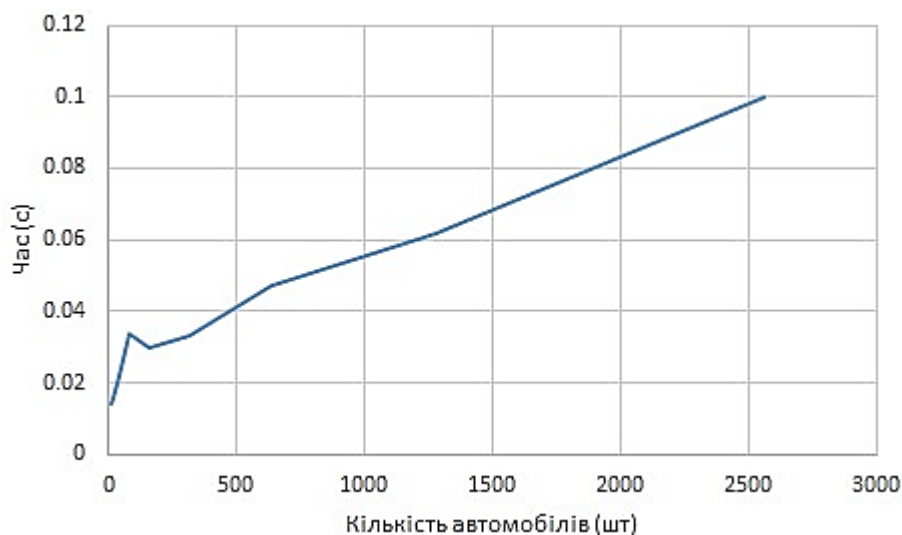


Рис. 3.2. Графік залежності часу роботи алгоритму від кількості автомобілів

На графіку видно, що час роботи алгоритму поступово збільшується. Однак, коли кількість наявних автомобілів збільшилася в 25 разів, робочий час збільшився в 5 разів. З урахуванням показника 2500 автомобілів алгоритм виконується за 0,1 секунди, що є дуже хорошим результатом. Ці результати вказують на те, що алгоритм відповідає критеріям масштабованості для систем рекомендацій, що вказує на те, що з часом пошук рекомендацій не займе багато часу.

Також було досліджено залежність середнього значення ступеня подібності вектору побажань користувачів і векторів рекомендованих автомобілів від кількості автомобілів, серед яких є рекомендації.

На рис. 3.3 наведено фрагмент графіка залежності середнього значення коефіцієнта подібності вектору атрибутів користувача та векторів атрибутів

рекомендованих елементів від загальної кількості векторів ознак автомобілів BMW у системі.

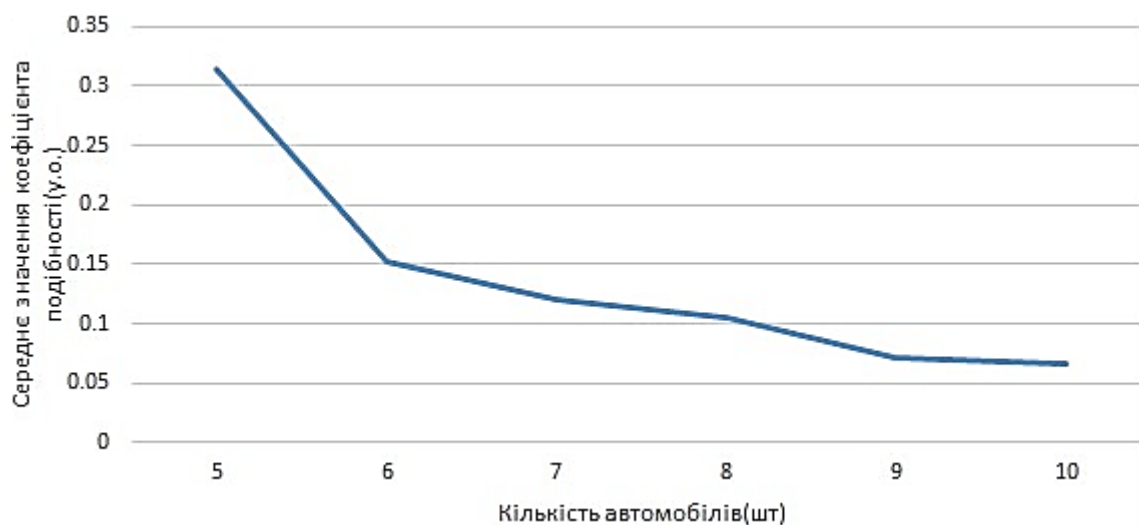


Рис. 3.2. Фрагмент графіку залежності середнього значення коефіцієнта подібності від загальної кількості автомобілів марки BMW в системі

Як видно з фрагмента графіка, зі збільшенням кількості доступних автомобілів середня відстань від вектору бажань клієнта до векторів ознак автомобіля зменшується, а це означає, що зі збільшенням кількості автомобілів у системі користувач отримає список рекомендацій, який включатиме найбільш підходящі автомобілі. .

Також було досліджено залежність середнього значення коефіцієнта подібності між вектором атрибутів користувача та векторами атрибутів рекомендованих елементів від кількості атрибутів у векторах ознак автомобіля.

Графік даної залежності зображено на рис. 3.4.

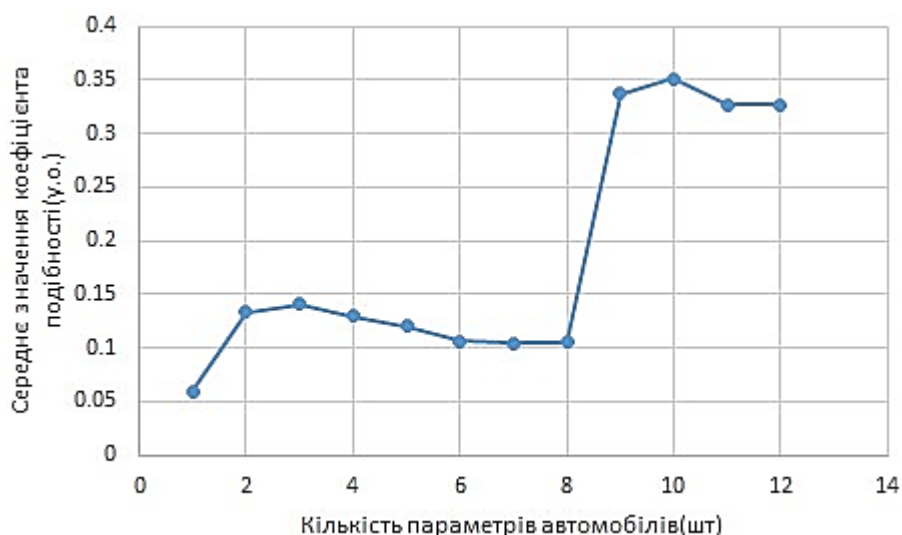


Рис. 3.3. Графік залежності середньої середнього значення коефіцієнта подібності від кількості атрибутів в векторах характеристик автомобілів

На графіку видно, що після восьми параметрів спостерігається значний стрибок середньої відстані, а це означає, що з дев'ятьма атрибутами у векторах автомобілів були знайдені автомобілі, які будуть рекомендовані клієнту, будуть більш різними і менше відповідатимуть його потребам. Тому під час пошуку рекомендацій використовуємо вісім параметрів у векторах атрибутів характеристик автомобіля.

Розроблений алгоритм можна використовувати для пошуку рекомендацій, тому що він стійкий до зростання обсягу даних, тому відповідає індексу масштабованості для рекомендаційних систем, і в разі збільшення кількості автомобілів в системі користувач буде отримати список рекомендацій, який складатиметься з найбільш підходящих автомобілів.

Висновки до розділу 3

Про моделі та методи пошуку рекомендацій сформульовано змістовну та математичну постановку задачі пошуку рекомендацій, розглянуто існуючі методи пошуку рекомендацій, розроблено та описано алгоритм, розроблено блок-схему алгоритму, наведено приклад роботи алгоритму та результати експериментальних досліджень. Під час огляду існуючих методик було

виявлено, що жодна публікація не висвітлює ситуацію, коли для пошуку рекомендацій необхідно, виходячи з побажань користувача, представити рекомендації серед існуючих об'єктів, тому обраний напрямок є важливим щодо дослідження та пошуку систем рекомендацій.

РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для створення фронтальної частини веб-додатку використано фреймворк Bootstrap [25] та бібліотеку JQuery на основі мови програмування JavaScript [26]. ASP.Net Core використовувався для розробки серверної частини веб-додатку на мові C# [27]. Оскільки серверна частина веб-програми створює записи в Dynamics CRM, для зв'язку з CRM використовувався API. Для створення HTTP-клієнта використовувалися Nuget [26] і пакет Dyrrix [29]. В якості системи для роботи з клієнтськими даними, управління даними та обробки запитів була обрана Dynamics 365 ForSales. Для створення обробника подій використовувалася платформа .Net 4.6.2. Використання інтерфейсу IOrganisationService [28] було здійснено для виконання основних операцій з сутністю. Для зберігання даних було обрано Common Data Service (CDS) [31].

C# - це мова програмування, яка використовує компілятор для перетворення коду в код IL, який потім перетворюється на двійковий код. Такий підхід дозволяє знайти всі синтаксичні помилки вже на етапі побудови та компіляції проекту, що є перевагою в порівнянні з мовами, які використовують інтерпретатор. C# є однією з мов, що входять до складу платформи .Net. Основним інструментом, який використовують розробники .Net, є Microsoft Visual Studio. Ця IDE містить дуже велику кількість різноманітних функціональних можливостей: код налагодження, код відстеження, відстеження продуктивності та витрат на ресурси, публікація програм у Azure, додавання функцій відповідно до ваших потреб шляхом розробки розширень тощо. C# часто отримує оновлення від Microsoft, а розробники часто публікувати власні бібліотеки як пакети Nuget, які можна безкоштовно використовувати навіть у комерційних цілях. C# також використовується в різних областях програмування: веб-розробка, розробка

мобільних додатків, розробка настільних додатків, розробка хмарних додатків, розробка ігор.

JavaScript - це динамічна мова програмування, яка не є строго типізованою, що означає, що об'єкт може мати один тип, але після операції присвоєння він може стати чимось зовсім іншим. Крім того, JavaScript не може працювати з пам'яттю, але основною метою мови є робота з веб-сайтами та браузером, тому працювати з пам'яттю не потрібно. Використовуючи JavaScript, сценарії в основному розробляються для динамічної зміни тегів HTML, анімації елементів сторінки, зміни обов'язкових полів, обчислення конкретних значень на основі даних сторінки та виклику методів API. JavaScript використовує інтерпретатор, тому, розробляючи сценарії, слід ще раз перевірити розробки та переконатися, що добре провели тестування розробки, оскільки помилки виникатимуть лише під час виконання певного рядка коду. Виклик методів API можна реалізувати через Ajax, що гарантує оновлення даних без необхідності перезавантажувати сторінку.

jQuery - це бібліотека, вбудована в JavaScript. Ця бібліотека полегшує та прискорює написання сценаріїв завдяки великій кількості функціональних можливостей у її базових реалізаціях. Бібліотека також значно полегшує створення анімацій завдяки тому, що вбудовані функції підтримуються більшістю браузерів, усуваючи необхідність змінювати сценарій для роботи в кожному браузері окремо. jQuery також дозволяє легко писати обробники подій на сторінках, динамічно змінювати теги HTML і писати запити Ajax.

Bootstrap - це фреймворк, який використовує HTML, CSS, JavaScript. Цей фреймворк містить велику кількість вбудованих класів CSS, функцій javascript і HTML-елементів, які разом дозволяють прискорити процес створення сайту за допомогою готових елементів. Великою перевагою використання цього фреймворку є те, що всі елементи адаптивні та працюють у більшості браузерів. Більш того, найчастіше фреймворк, крім елементів кастомізації, використовується для побудови сіток. Цей фреймворк надає дуже гнучкі

інструменти для їх створення, тому він може задовольнити потреби будь-якого розробника. Щоб використовувати цей фреймворк, просто завантажте компоненти, помістіть їх у потрібну папку та посилайте на сайт або можете скористатися посиланнями на онлайн-версії цих бібліотек, але в цьому випадку, якщо сервери, що містять ці бібліотеки, недоступні, функціональність за допомогою Bootstrap не працюватиме.

Dyrix - це пакет NuGet для створення HTTP-клієнта для взаємодії з Dynamics CRM. Щоб підключитися до CRM через API, необхідно спочатку отримати маркер, а потім надсилати запити, використовуючи його. Максимальний термін дії токена становить 1 годину. Перевага використання цього пакета полягає в тому, що він автоматично перевіряє дійсність маркера, і якщо термін дії маркера закінчується, пакет автоматично запитує новий і використовує його для наступних запитів.

NuGet - це менеджер пакетів для .NET. За допомогою вбудованих інструментів NuGet можна публікувати свої проекти в галереї пакетів, а також завантажувати та використовувати проекти інших людей. Галерея NuGet - це спільне сховище, яке містить усі пакети та використовується розробниками пакетів і людьми, які використовують ці пакети.

Dynamics 365 - це сімейство продуктів, що включає рішення ERP і CRM. При розробці системи рекомендацій була взята на озброєння одна з різновидів рішень Dynamics CRM - Dynamics 365 for Sales, як інтегрована система управління даними про автомобіль і клієнта, управління робочими процесами з клієнтами та алгоритм пошуку рекомендацій в Sales. На рис. 4.1 показано структуру сімейства продуктів Dynamics 365.

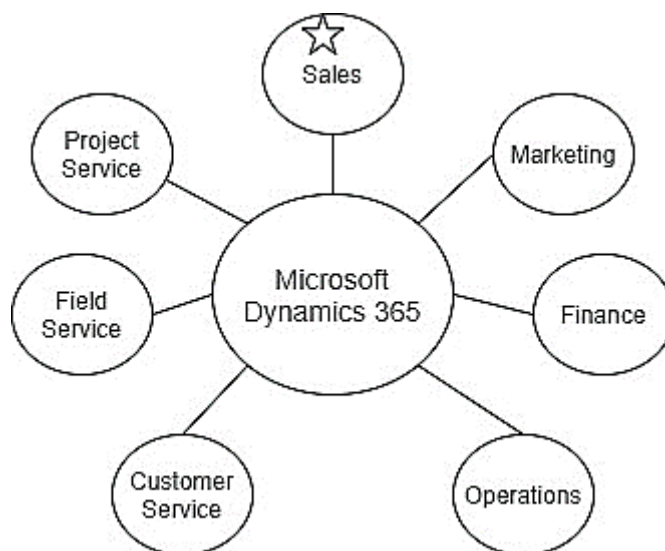


Рис. 4.1. Структура сімейства продуктів Dynamics 365

Microsoft Dynamics 365 for Sales - це модуль Dynamics 365, який відповідає за продажі та дозволяє збільшувати обсяги продажів за рахунок зберігання даних клієнтів і вибудовування індивідуального підходу до кожного клієнта на основі наявних даних. Microsoft Dynamics 365 for Sales дозволяє відстежувати хід роботи з клієнтами, отримувати різноманітну статистику на основі даних і використовувати дані клієнтів у маркетингових цілях.

Інтерфейс `IOrganizationService` використовується розробниками CRM, які працюють із Dynamics 365. Цей інтерфейс містить набір методів для роботи з настроюваними та системними сутностями. Інтерфейс дозволяє виконувати такі операції, як створення, оновлення, видалення, отримання одного запису, отримання кількох записів, виконання системних запитів, виклик користувацьких процесів, а також дозволяє отримувати метадані сутності.

CDS (Common Data Service) - сервіс, який дозволяє працювати з даними. Використовуючи цей сервіс, розробники працюють не з таблицями в базах даних, а з об'єктами, які є сутностями, які можна легко налаштувати через інтерфейс. Сутності в CDS перетворюються на записи таблиці бази даних. Перевагою CDS є те, що його можна використовувати в різних програмах Microsoft.

4.2 Архітектура програмного забезпечення

4.2.1. Структурна схема класів

Розглянемо діаграму класів для плагіна, що спрацьовує на оновленні записів сутності клієнт (Lead). Діаграму класів для плагіна зображено на рис. 4.2.

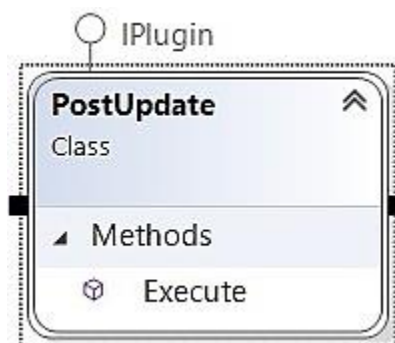


Рис. 4.2. Структурна схема класів плагіна

Плагін представлений класом PostUpdate. Цей клас реалізує інтерфейс IPlugin і містить метод Execute. Інтерфейс IPlugin – це інтерфейс, який використовується всіма плагінами, опублікованими або опублікованими в Dynamics CRM. Використання цього інтерфейсу гарантує, що кожен плагін включатиме метод Execute. Метод Execute містить основну логіку плагіна, а також отримує контекст виклику, об'єкт, що містить усі зміни, і службу відстеження. Для роботи з об'єктами були розроблені додаткові класи. Класи розширення були розроблені для класу Car, Entity, інтерфейсу IEnumerable. Діаграма класів розширення показана на рис. 4.3.

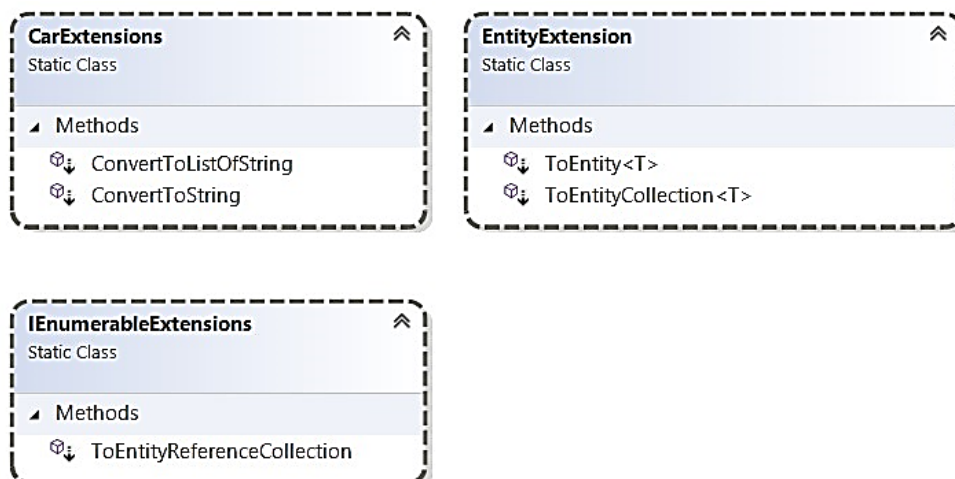


Рис. 4.3. Структурна схема класів розширень

Класи розширення приймають як свій перший параметр екземпляр класу, який буде використано, і додаткові параметри можуть використовуватися за потреби. Клас `CarExtensions` містить 2 методи:

1. `ConvertToListOfString` – перетворює список об'єктів типу `Car` в список об'єктів, що містять текстовий опис характеристик об'єктів;
2. `ConvertToString` – перетворює об'єкт типу `Car` в текстовий опис характеристик об'єкта.

Клас `EntityExtension` містить 2 методи:

1. `ToEntity<T>` – загальний метод, що перетворює об'єкт `Entity` в об'єкт типу `EntityProvider`, що успадковується від моделі певного класу, що в свою чергу успадковується від `EntityBase` класу;
2. `ToEntityCollection<T>` – загальний метод, що перетворює `EntityCollection` в список об'єктів `EntityProvider`-ів за допомогою методу `ToEntity<T>`.

Клас `IEnumerableExtensions` містить 1 метод:

1. `ToEntityReferenceCollection` – загальний метод, що перетворює `IEnumerable<EntityReference>` в `EntityReferenceCollection`.

Для роботи з `IOrganizationService` був обраний підхід `EarlyBound`. Це означає, що спочатку створюються моделі на основі відповідних полів у `Dynamics CRM`, а потім ці моделі використовуються для роботи з `CRM`. Діаграма класів для взаємодії з `CRM` зображено на рис. 4.4.

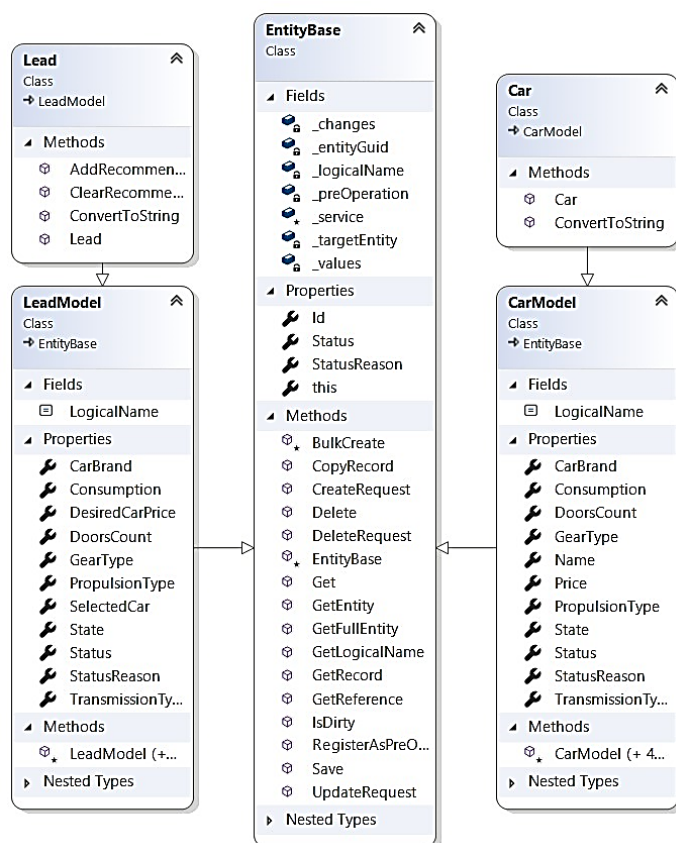


Рис. 4.4. Структурна схема класів для взаємодії з CRM

На діаграмі показано 2 класи EntityProvider Lead і Car. Класи EntityProvider успадковують від моделей EarlyBound LeadModel і CarModel відповідно. Клас Lead містить 4 методи:

1. AddRecommendations – асоціює знайдені записи рекомендацій об'єктів типу Car з записом типу Lead в CRM за зв'язком багато до багатьох;
2. ClearRecommendations – видаляє зв'язки між знайденими записами рекомендацій об'єктів типу Car з записом типу Lead в CRM за зв'язком багато до багатьох;
3. ConvertToString – перетворює об'єкт типу Lead в текстовий опис характеристик об'єкта;
4. Lead – конструктор класу.

Клас Car містить 2 методи:

1. Car – конструктор класу;

2. `ConvertToString` – перетворює об'єкт типу `Car` в текстовий опис характеристик об'єкта.

Клас `LeadModel` є моделлю `EarlyBound` і містить необхідні параметри з CRM у вигляді властивостей.

Клас `CarModel` є моделлю `EarlyBound` і містить необхідні параметри з CRM у формі властивостей.

Класи `LeadModel` і `CarModel` успадковують клас `EntityBase`.

`EntityBase` - базовий клас, від якого успадковуються всі моделі. Містить 16 методів:

1. `IsDirty` – повертає значення `true` чи `false` в залежності від того чи змінювалась модель з моменту створення отримання;
2. `RegisterAsPreOperation` – реєструє об'єкт, як `target` для використання в `PreOperation` плагінах;
3. `Save` – використовується для збереження змін в CRM для сутності;
4. `Delete` – видаляє запис сутності в CRM за допомогою інтерфейсу `IOrganizationService`;
5. `Get` – дозволяє отримати один або декілька атрибутів для поточної сутності;
6. `GetRecord` – дозволяє отримати запис сутності з CRM;
7. `CopyRecord` – дозволяє отримати об'єкт що є копією того об'єкта для якого було викликано метод;
8. `DeleteRequest` – видаляє запис сутності в CRM за допомогою відповідного запиту;
9. `UpdateRequest` – оновлює запис сутності в CRM за допомогою відповідного запиту;
10. `CreateRequest` – створює запис сутності в CRM за допомогою відповідного запиту;

11. `GetFullEntity` – отримує об'єкт `Entity` з `EarlyBound` моделі;
12. `GetReference` – отримує об'єкт `EntityReference` з `EarlyBound` моделі;
13. `GetLogicalName` – отримує логічне ім'я сутності в CRM з `EarlyBound` моделі;
14. `BulkCreate` – дозволяє створювати багато записів за один запит до CRM;
15. `CreateRequest` – повертає об'єкт запиту створення. Використовується в оновленні багатьох записів в CRM за один запит;
16. `Entitybase` – конструктор класу.

Для пошуку рекомендацій у плагіні було створено клас `RecommendationSearchService`. Об'єкт цього класу створюється кожного разу, коли плагін викликається. Цей клас містить список автомобілів, отриманий із наявних записів у CRM, спеціальний об'єкт, мінімальне та максимальне значення для нормалізації. Також були створені додаткові моделі для пошуку рекомендацій. Блок-схема класів, які використовуються для пошуку рекомендацій, наведена на рис. 4.5.

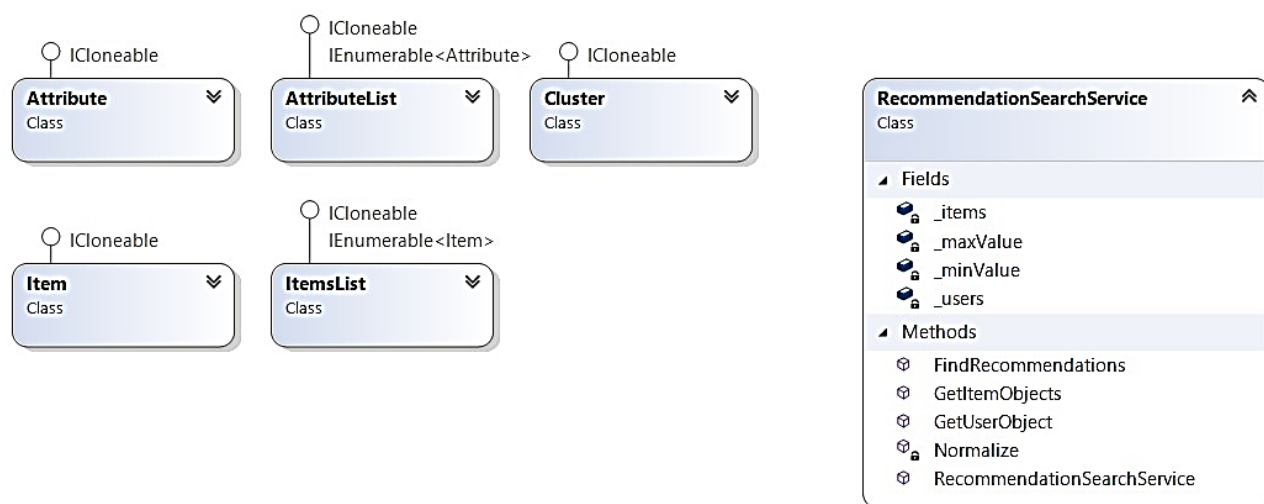


Рис. 4.5. Структурна схема класів для знаходження рекомендацій

Клас `Item` - це клас елементів користувача або автомобіля, клас `ItemList` - це список класів користувачів або автомобілів, `Attribute` - це клас із певною функцією або бажанням, `AttributeList` - це список класів функцій або бажань, `Cluster` - це клас група предметів.

RecommendationSearchService містить методи для отримання об'єктів списку автомобілів і об'єктів користувача, метод пошуку рекомендацій, конструктор класу та метод нормалізації індексації елементів.

Для перетворення значень, отриманих від системи CRM, був розроблений інтерфейс IConverter, який забезпечить наявність у всіх класів, що реалізують цей інтерфейс, методу Convert. Класи Converter реалізують інтерфейс IConverter і мають один метод Convert. Також був створений клас ConversionService, що містить метод Convert, який приймає назву класу конвертера як параметр, ініціалізує інтерфейс IConverter і викликає метод Convert відповідного конвертера. Діаграма класів для перетворення даних з CRM в текстові дані наведена на рис. 4.6.

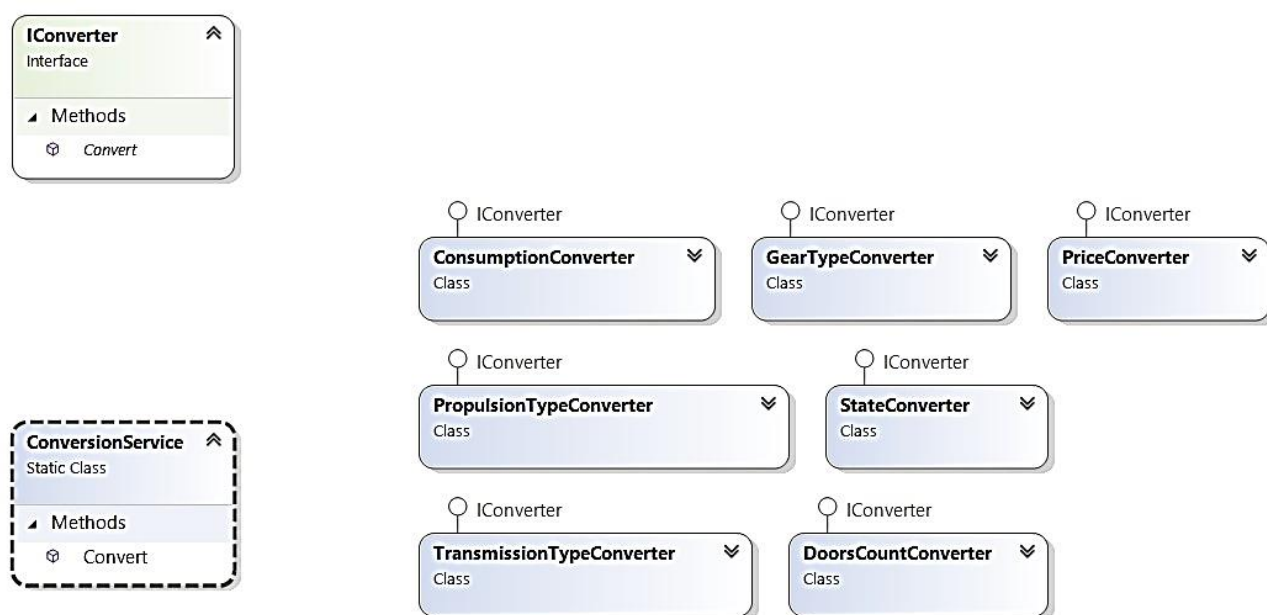


Рис. 4.6. Структурна схема класів для перетворення даних з CRM в текстові дані

4.2.2 Структурна схема послідовності

Наведемо структурну схему послідовності (рис. 4.7).

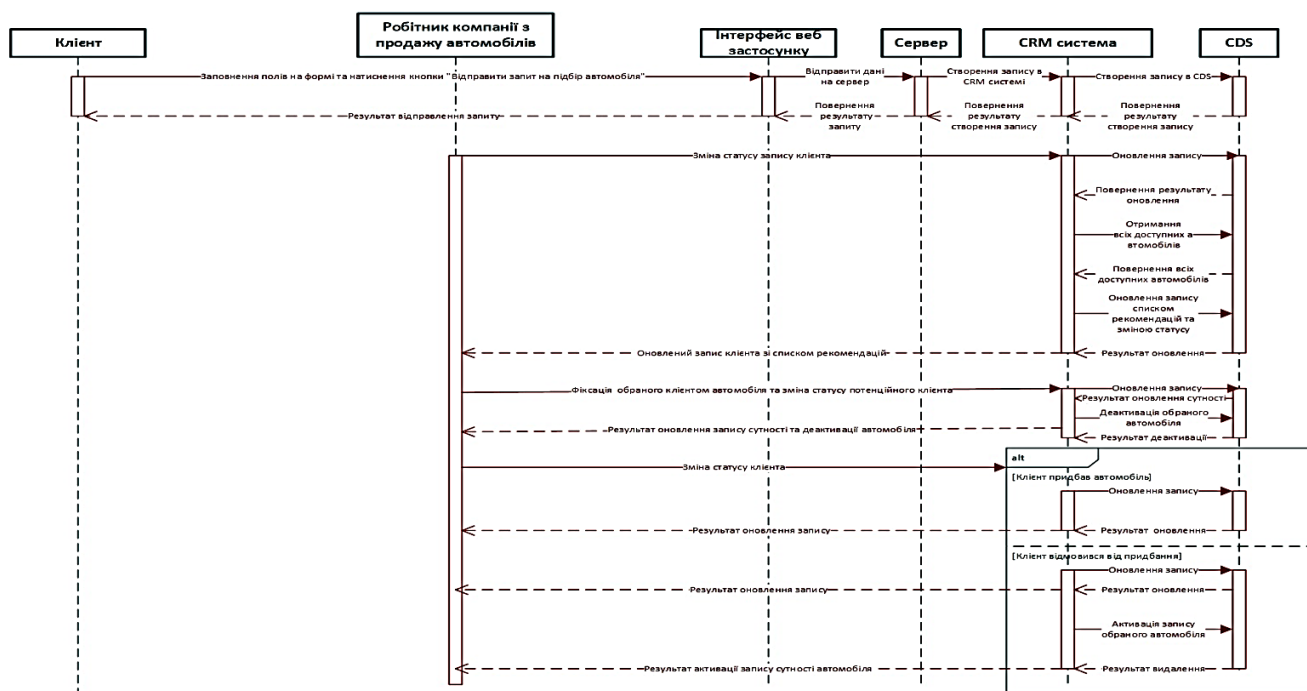


Рис. 4.7. Структурна схема послідовності

Клієнт працює тільки з інтерфейсом веб-додатку. Інтерфейс передає дані на сервер, який, у свою чергу, передає дані в систему CRM, а також в CDS. Якщо дані були успішно збережені, клієнт отримує сповіщення про те, що його запит успішно прийнято.

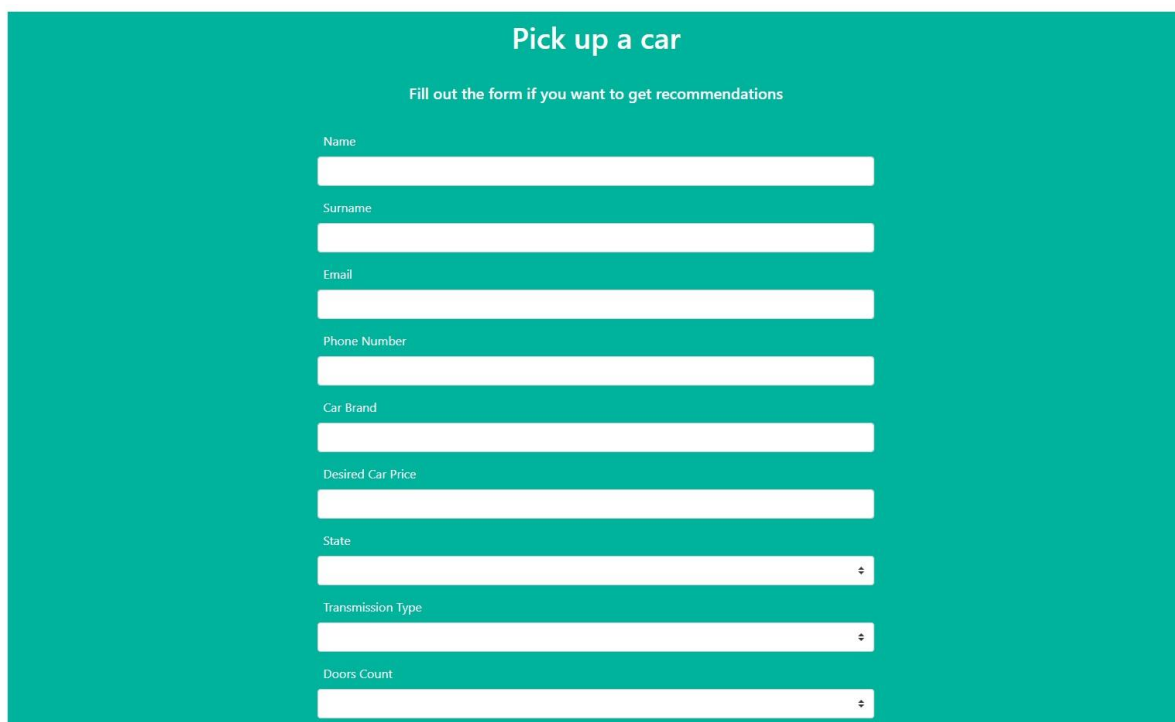
Якщо розглядати співробітника компанії з продажу автомобілів, то він взагалі працює не з веб-додатком, а тільки з CRM-системою. Коли співробітник працює з CRM-системою, завжди задіяні тільки два компоненти: CRM-система, з якою працює співробітник, і CDS. Після кожної операції співробітник компанії отримує повідомлення про помилку, якщо така подія сталася, або результат операції.

4.3. Інструкція користувача

4.3.1 Інструкція клієнта

Для отримання рекомендацій користувачеві необхідно зайти в онлайн-заявку та заповнити форму заявки на підбір автомобіля.

На рис. 4.8 зображено форму заявки на підбір автомобіля.

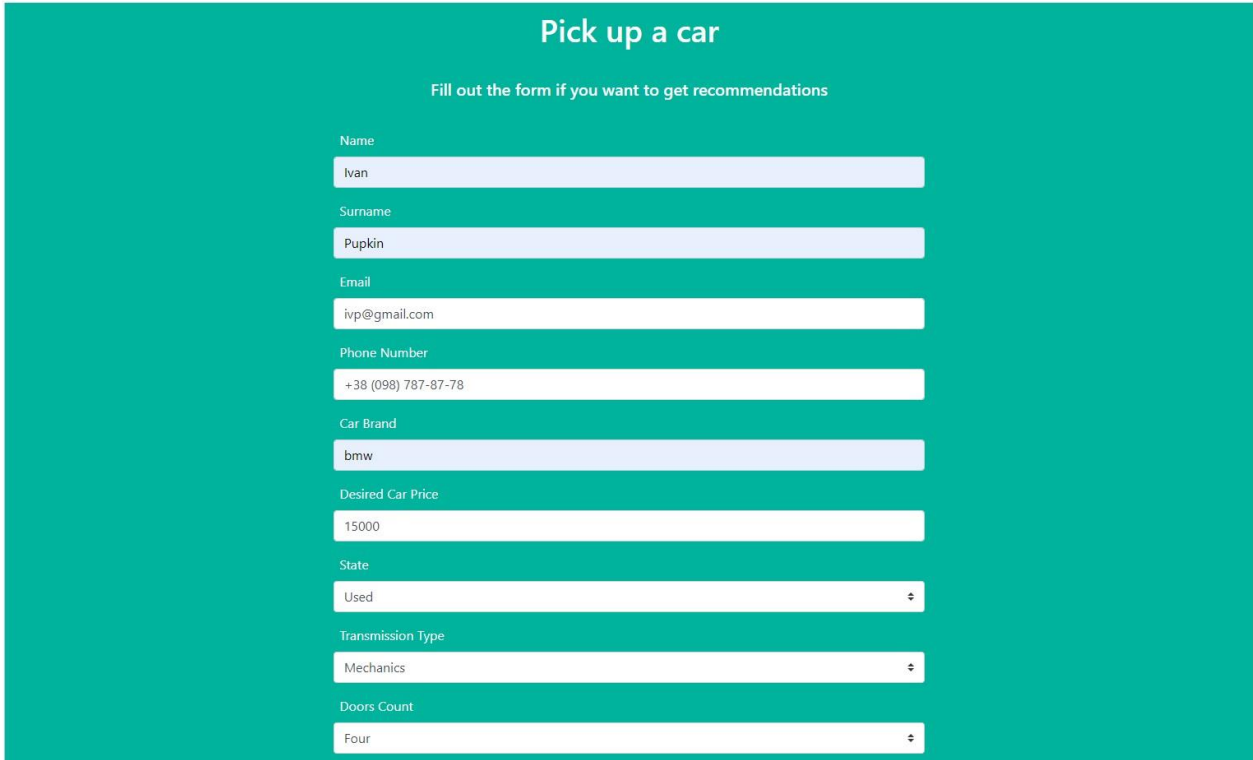


The image shows a web form titled "Pick up a car" on a teal background. Below the title is the instruction "Fill out the form if you want to get recommendations". The form consists of several input fields: "Name", "Surname", "Email", "Phone Number", "Car Brand", "Desired Car Price", "State", "Transmission Type", and "Doors Count". Each field is represented by a white rectangular box with a small downward arrow on the right side, indicating they are dropdown menus.

Рис. 4.8. Форма заявки на підбір автомобіля

Відкривши форму, клієнт повинен заповнити особисті дані (ім'я, прізвище, електронну адресу), побажання щодо характеристик автомобіля (марка автомобіля, бажана ціна, кількість дверей, тип приводу, тип приводу, обсяг споживання, стан, тип кузова) і натиснути прапорець, тим самим підтверджуючи згоду на обробку персональних даних.

На рис. 4.8 та 4.9 зображено приклад заповнення форми заявки на підбір автомобіля.



Pick up a car

Fill out the form if you want to get recommendations

Name
Ivan

Surname
Pupkin

Email
ivp@gmail.com

Phone Number
+38 (098) 787-87-78

Car Brand
bmw

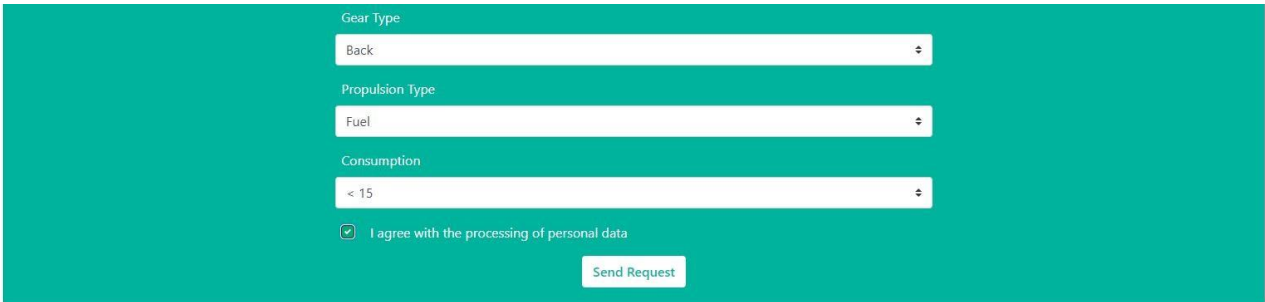
Desired Car Price
15000

State
Used

Transmission Type
Mechanics

Doors Count
Four

Рис. 4.9. Приклад заповнення форми заявки на придбання автомобіля



Gear Type
Back

Propulsion Type
Fuel

Consumption
< 15

I agree with the processing of personal data

Send Request



Рис. 4.10. Приклад заповнення форми заявки на придбання автомобіля

Після заповнення форми Замовнику необхідно натиснути кнопку «Надіслати запит». Після успішного надсилання запиту клієнт повинен побачити, що форма зникла та з'явиться повідомлення про те, що представник зв'яжеться з клієнтом.

На рис. 4.11 зображено зміну сторінки та відображення повідомлення про те, що заявка прийнято.

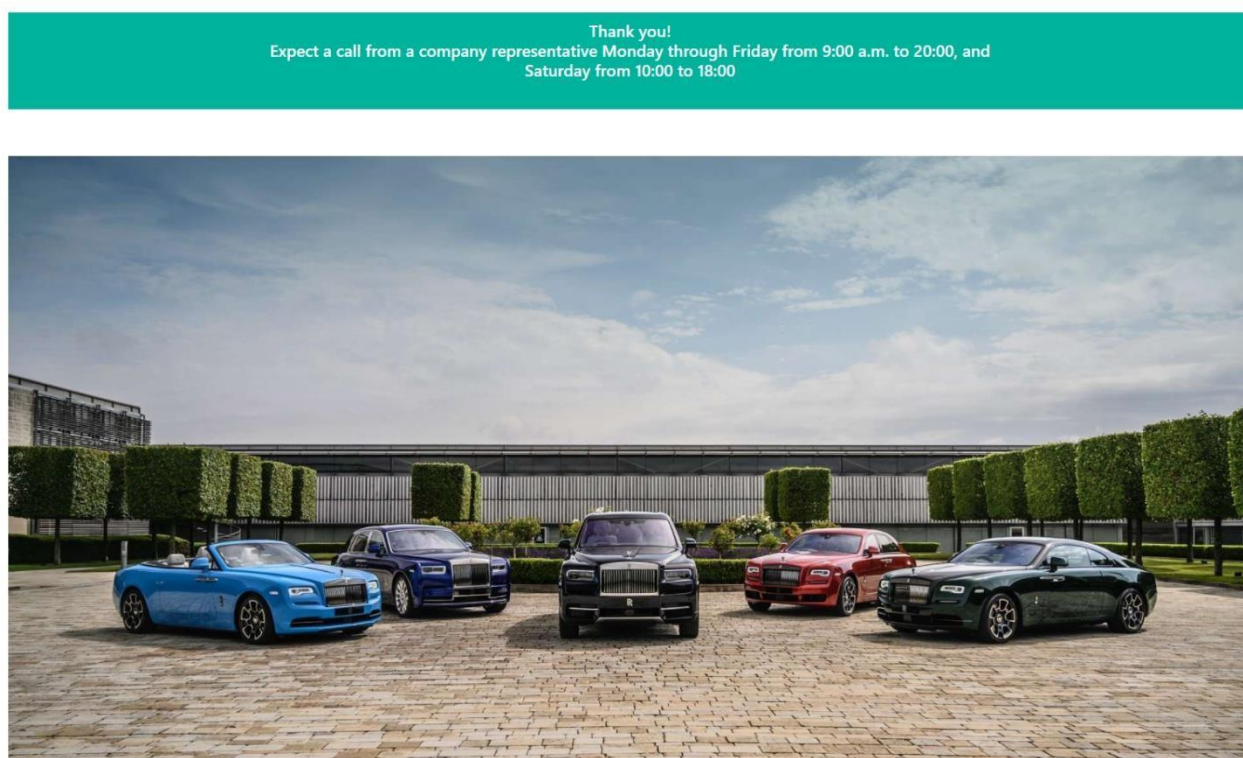


Рис. 4.11. Відображення повідомлення про прийняття заявки

4.3.2 Інструкція робітника компанії з продажу автомобілів

Щоб почати роботу з CRM, співробітник компанії з продажу автомобілів повинен авторизуватися. Для аутентифікації введіть свій логін і пароль і натисніть кнопку «SignIn».

На рис. 4.12 та 4.13 зображено форми вводу логіна та паролю відповідно.

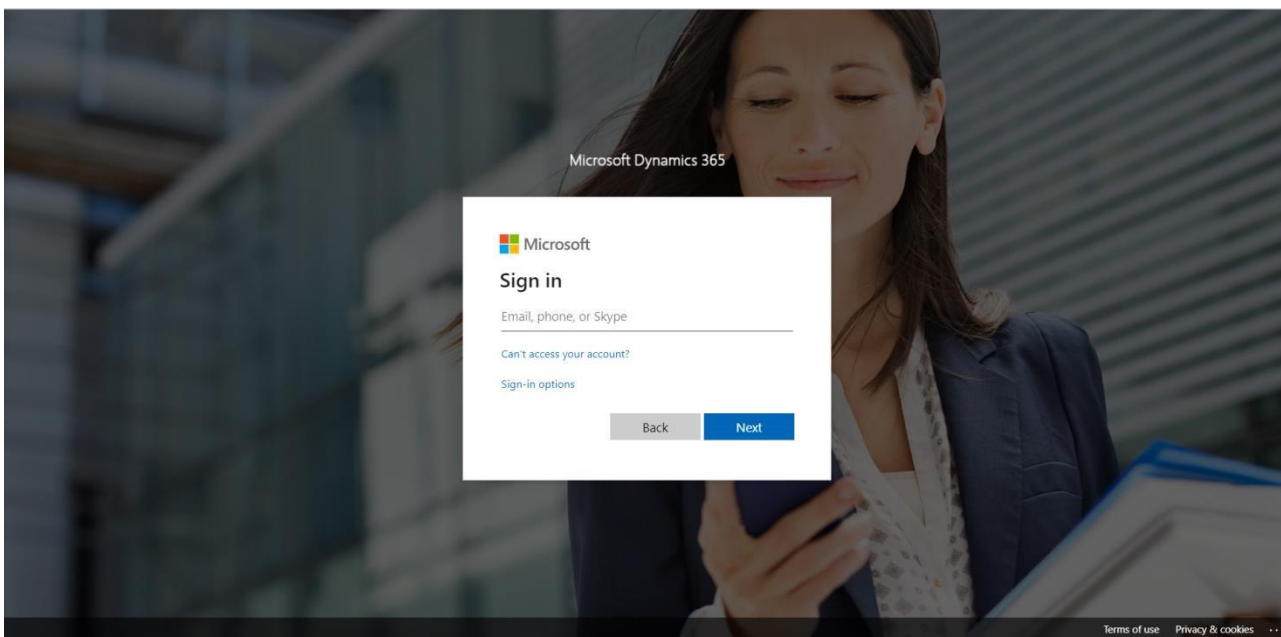


Рис. 4.12. Форма вводу логіна

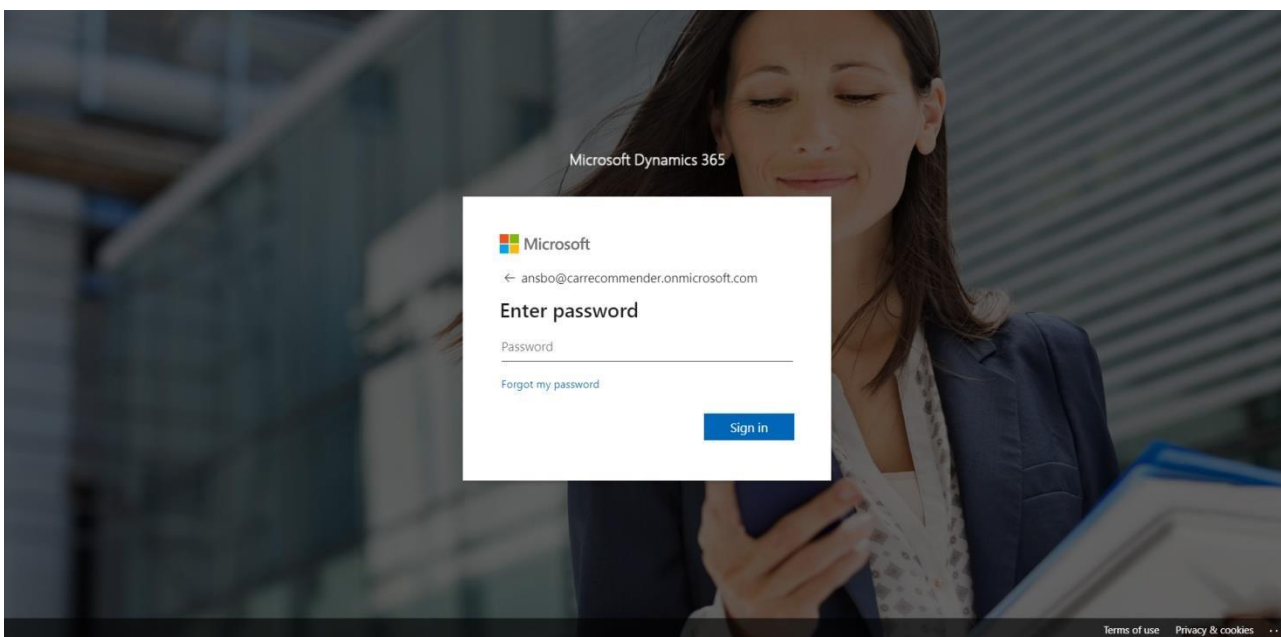


Рис 4.13. Форма вводу паролю та входу в систему

Після авторизації співробітник компанії переходить на головну сторінку CRM-системи, де може вибрати потрібний пункт меню, перейти до налаштувань і переглянути графік кількості залучених клієнтів з різних джерел.

На рис. 4.14 зображено головну сторінку CRM системи.

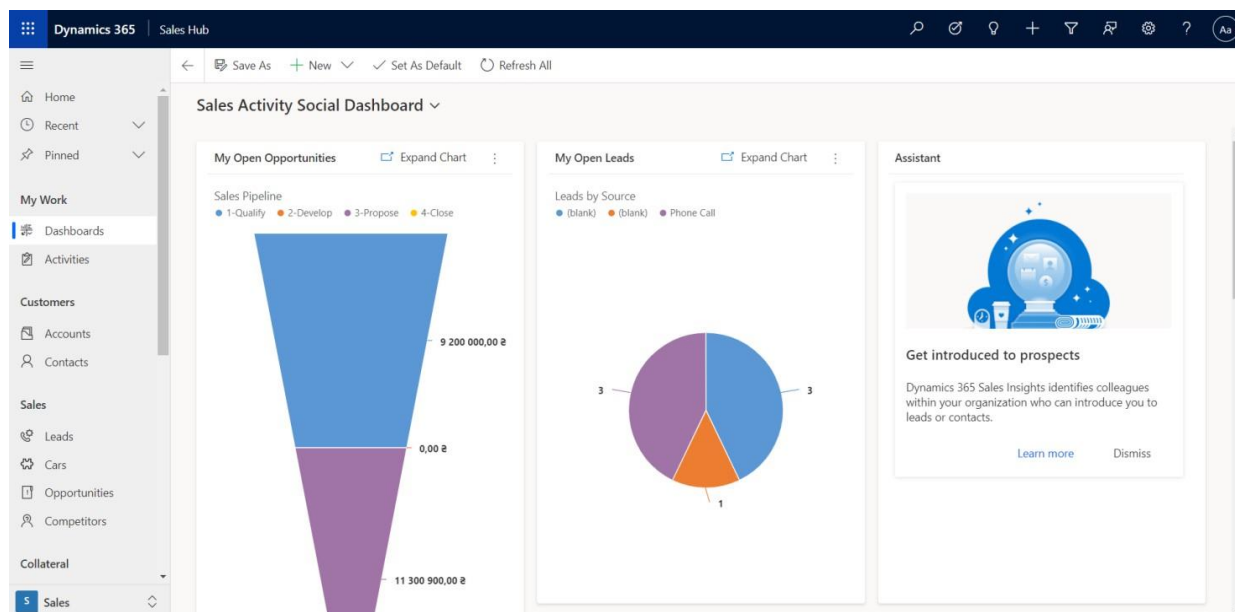


Рис. 4.1. Головна сторінка CRM системи

Якщо клієнт дзвонить в компанію і хоче отримати рекомендації без використання онлайн-додатку, співробітник компанії з продажу автомобілів повинен перейти до списку клієнтів, натиснувши «Leads» в меню зліва. Список клієнтів наведено на рис. 4.15.

Name	Topic	Status Reason	Created On
Alex Wu	Expressed interest in A. Datum X line of print...	New	21.01.2017 0:52
Anton Sboryk	Phone call 28/10	RecommendationsFound	28.10.2020 20:13
Brian LaMee	Interested in online only store	New	25.10.2020 3:22
George Sullivan	Store is expanding - send new literature	New	25.10.2020 3:22
Kim Abercrombie	Likes our products	New	25.10.2020 3:22
Vlada Chernenko	Phone call 31/10/20 20:30	RecommendationsFound	31.10.2020 20:28
Yurii Sboryk	Phone 31/10	CarSelected	31.10.2020 18:59

Рис. 4.15. Відображення списку клієнтів

Для того, щоб зареєструвати нове звернення робітник натискає на кнопку «New» з верхнього меню. Форма створення нового клієнта зображена на рис. 4.16.

The screenshot shows the Dynamics 365 Sales Hub interface for creating a new lead. The left sidebar contains navigation options like Dashboards, Activities, Customers, Sales, and Collateral. The main area is titled 'New Lead' and includes a 'Summary' tab. The 'Contact' section has several input fields, most of which are empty. The 'Timeline' section shows a message: 'Almost there. Select Save to see your timeline.' The top right corner indicates the lead source as 'Ansbo ansbo' and the status as 'New'.

Рис. 4.16. Форма створення нового клієнта

Співробітник повинен надати персональні дані Замовника, такі як ім'я, прізвище, номер телефону, адреса електронної пошти та вказати тему запиту. Приклад заповнення форми із персональними даними та джерелом запиту наведено на рис. 4.17.

This screenshot shows the same 'New Lead' form, but now it is populated with data. The 'Contact' section contains the following information: Topic is 'Phone Call 07/11/20', First Name is 'Petr', Last Name is 'Petrov', Business Phone is '380577575542', Mobile Phone is '380577575542', Email is 'pp@ukr.net', and Is Consent Given is 'Yes'. The 'Timeline' section remains empty with the 'Almost there' message. The top right corner now shows the lead source as 'Phone Call', the status as 'Warm', and the owner as 'Ansbo ansbo'. The bottom right corner indicates 'unsaved changes'.

Рис. 4.17. Приклад заповнення форми персональними даними та вибору джерела звернення

Після заповнення особистих даних співробітник повинен перейти на вкладку CarSelection. Форма для заповнення побажань клієнта та роботи з клієнтом при виборі автомобіля наведена на рис. 4.18.

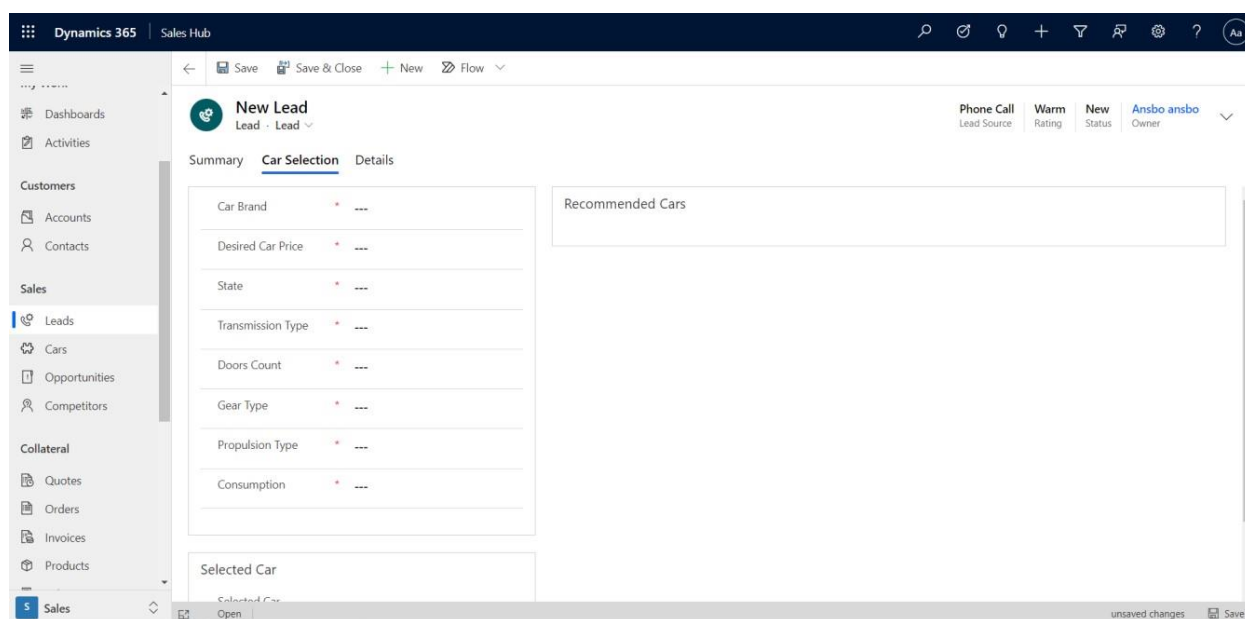
The screenshot shows the Dynamics 365 Sales Hub interface. The main window displays a 'New Lead' form with the 'Car Selection' tab selected. The form includes several input fields: 'Car Brand', 'Desired Car Price', 'State', 'Transmission Type', 'Doors Count', 'Gear Type', 'Propulsion Type', and 'Consumption'. To the right of these fields is a 'Recommended Cars' section. The left sidebar shows the navigation menu with 'Leads' and 'Cars' highlighted. The top navigation bar includes 'Dynamics 365 | Sales Hub' and various utility icons. The bottom status bar indicates 'unsaved changes' and a 'Save' button.

Рис. 4.18. Форма заповнення побажань клієнта та роботи з клієнтом по підборі автомобіля

Співробітник повинен заповнити побажання клієнта щодо автомобіля, а саме марку автомобіля, витрата палива, бажану вартість, кількість дверей, тип двигуна, тип приводу. Приклад заповнення форми запиту клієнта та роботи з клієнтом щодо вибору автомобіля наведено на рис. 4.19.

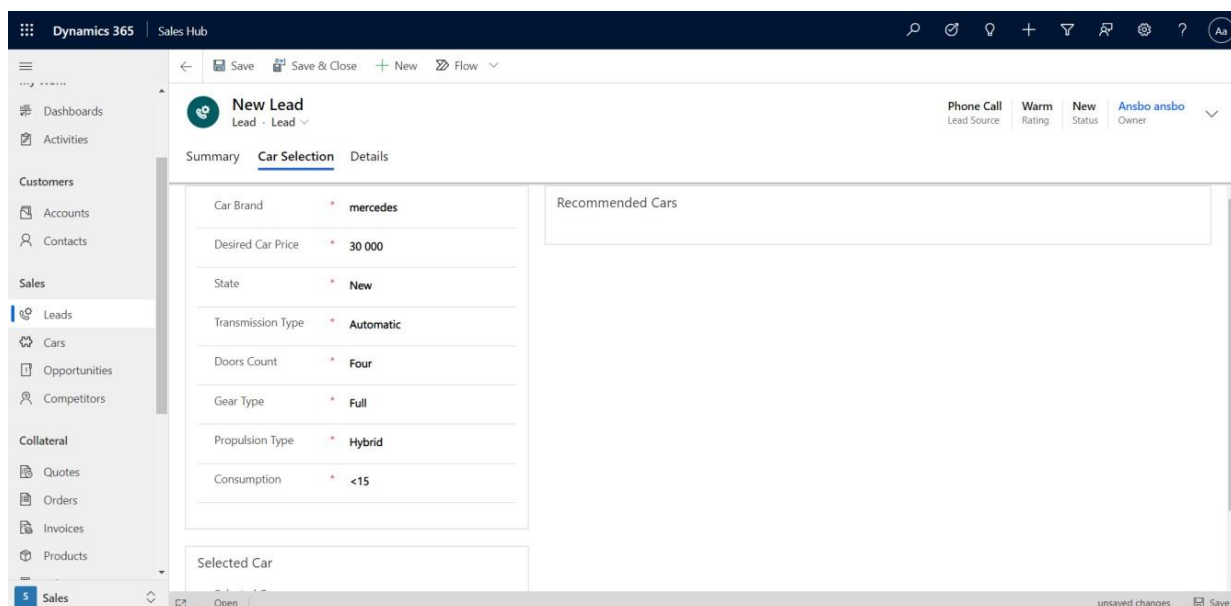
This screenshot is similar to the previous one, but the form fields are populated with example data. The 'Car Brand' field contains 'mercedes', 'Desired Car Price' is '30 000', 'State' is 'New', 'Transmission Type' is 'Automatic', 'Doors Count' is 'Four', 'Gear Type' is 'Full', 'Propulsion Type' is 'Hybrid', and 'Consumption' is '<15'. The 'Recommended Cars' section remains empty. The interface elements like the sidebar and top navigation bar are consistent with the previous image.

Рис. 4.19. Приклад заповнення форми побажань клієнта

Розглянемо список дій, які повинен виконати співробітник, якщо запит від клієнта надходить з веб-додатку. Співробітник повинен вибрати перегляд AllLeads і переглянути список клієнтів зі статусом New. Список усіх клієнтів показаний на рис. 4.20.

Name	Topic	Owner	Status Reason	Created On
Petr Petrov	Phone Call 07/11/20	Ansbo ansbo	New	07.11.2020 18:14
Ivan Pupkin	Web app request 07-Nov-20 18:10:14	integration user	New	07.11.2020 18:10
Vasya Ivanov	Web app request 05-Nov-20 20:48:23	integration user	RecommendationsFound	05.11.2020 20:48
Petr Petrov	Phone call 03/11/20 20:26	Ansbo ansbo	CarBought	03.11.2020 20:30
Semen Semenov	Web app request 03-Nov-20 20:10:52	integration user	CarBought	03.11.2020 20:10
Vasya Pupkin	Web app request 01-Nov-20 18:12:58	integration user	CarBought	01.11.2020 18:12
Vlada Chernenko	Phone call 31/10/20 20:30	Ansbo ansbo	No Longer Interested	31.10.2020 20:28
Yurii Sboryk	Phone 31/10	Ansbo ansbo	CarSelected	31.10.2020 18:59
Anton Sboryk	Phone call 28/10	Ansbo ansbo	No Longer Interested	28.10.2020 20:13
George Sullivan	Store is expanding - send new literature	Ansbo ansbo	New	25.10.2020 3:22
Kim Abercrombie	Likes our products	Ansbo ansbo	New	25.10.2020 3:22
Adrian Dumitrascu	Video Hardware Upgrade	David So (Sample Data)	CarBought	25.10.2020 3:22

Рис. 4.20 Відображення списку всіх клієнтів

Співробітник повинен вибрати клієнта зі статусом Новий і двічі клацнути лівою кнопкою миші у відповідному рядку. Після натискання відкривається форма клієнта, яка містить усі дані, які клієнт заповнив у веб-додатку, а джерело запиту вказується як «WebApp». На рис. 4.21 та 4.22 наведено приклади відображення персональних даних клієнта та запитів, виконаних клієнтом у веб-додатку.

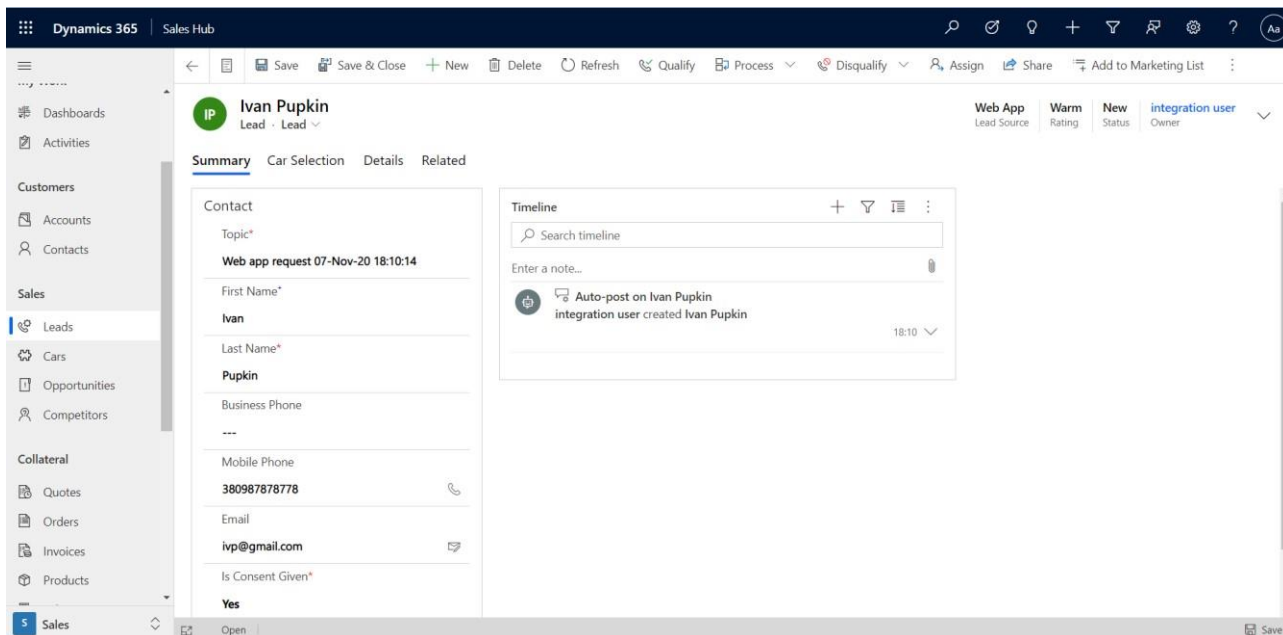


Рис. 4.21. Відображення персональних даних клієнта

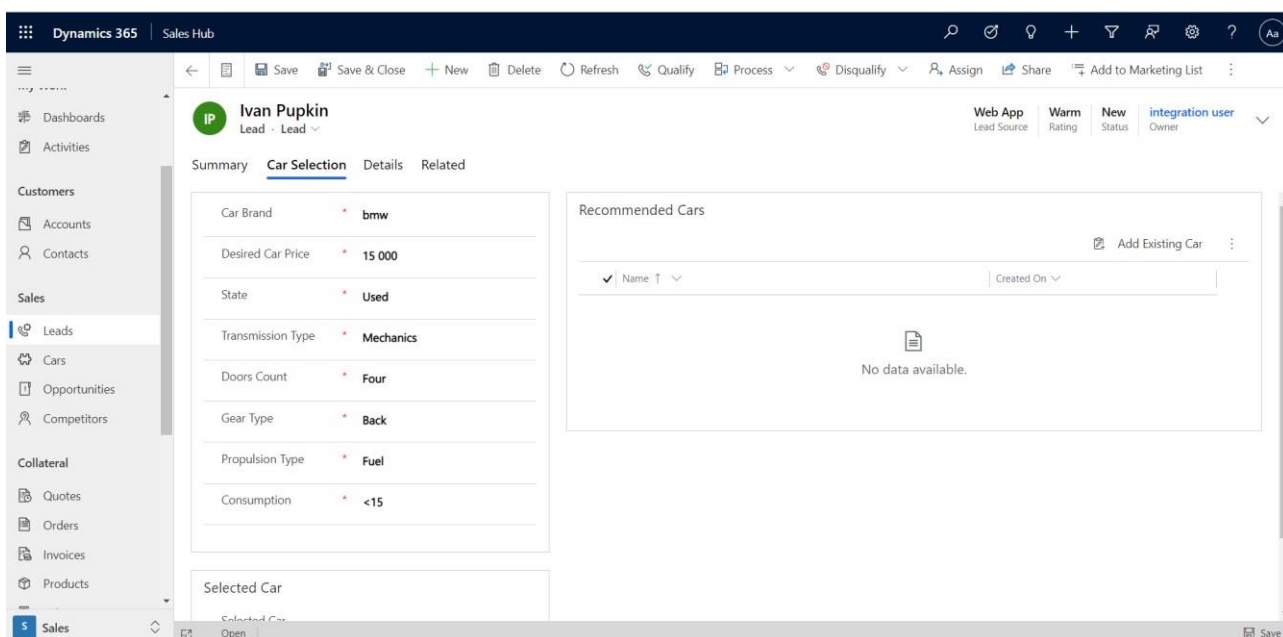


Рис. 4.22. Відображення побажань клієнта

Процес роботи з клієнтом однаковий для запитів, що надходять з веб-додатку та по телефону. Розглянемо цей процес для запиту клієнта, що надходить із веб-додатку. Щоб розпочати роботу з клієнтом, співробітник повинен змінити статус запису з Новий на Розпочато. Зміна стану запису та збереження запису видно на рис. 4.23 та 4.24.

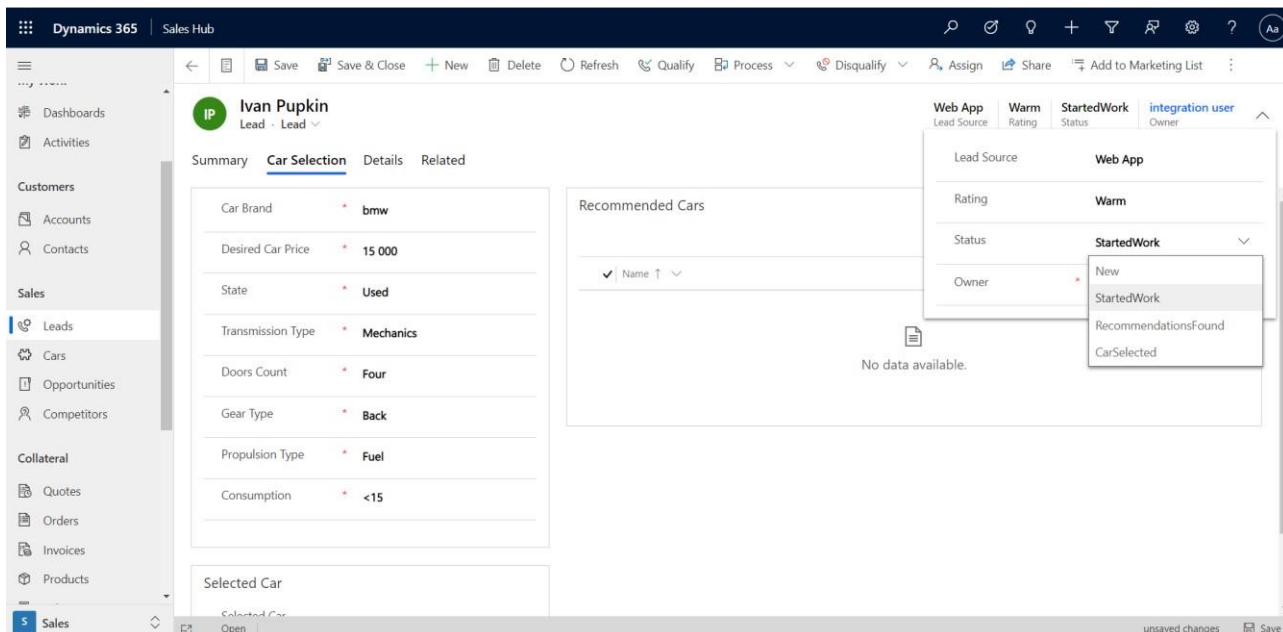


Рис. 4.23. Зміна статусу запису

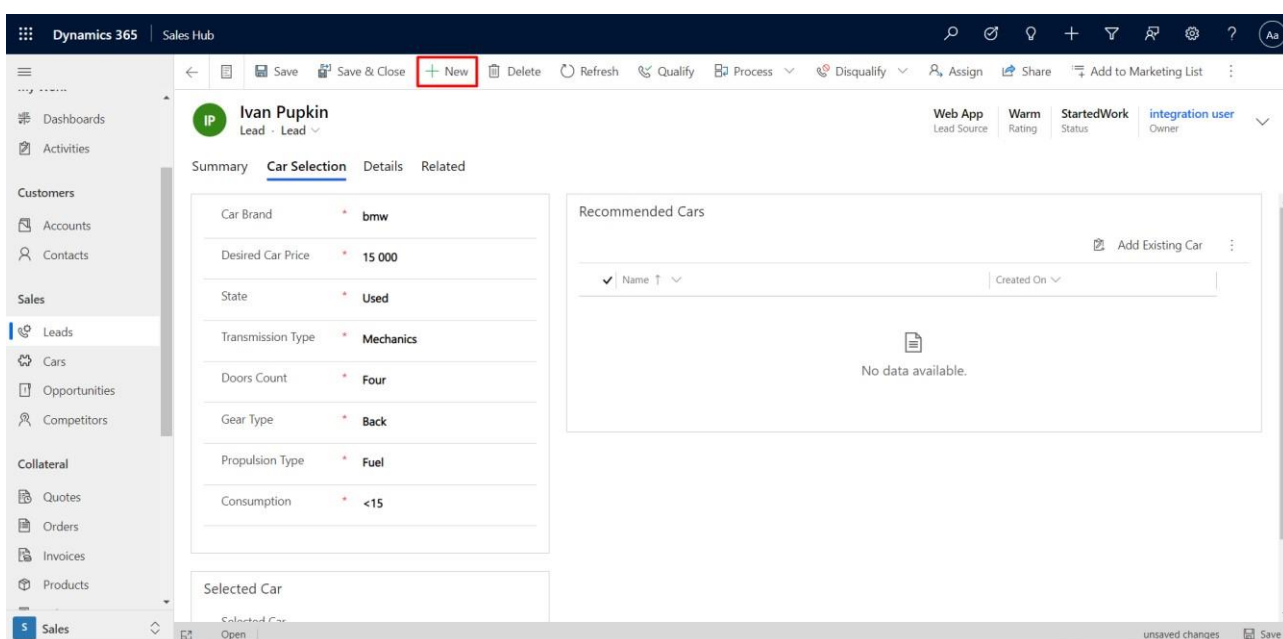


Рис. 4.24. Збереження запису

Після зміни статусу запису на StartedWork система шукає рекомендовані автомобілі та відображає їх як пов'язані записи у формі в таблиці Recommended Cars. Також система автоматично змінює статус запису на "RecommendationsFound".

Відображення списку рекомендацій зображено на рис. 4.25.

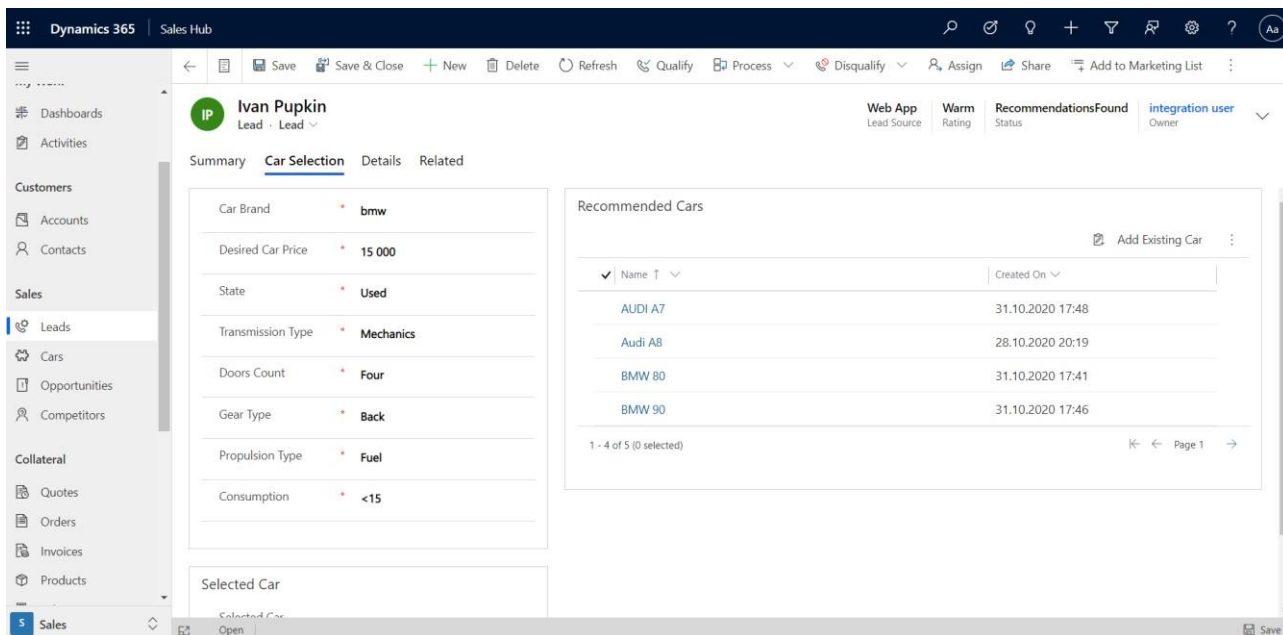


Рис. 4.25. Відображення списку рекомендацій

Після того, як рекомендація знайдена, співробітник компанії з продажу автомобілів спілкується з клієнтом за допомогою мобільного телефону або електронної адреси, яку клієнт надав під час запиту рекомендації. Після того, як клієнт вибере автомобіль зі списку рекомендованих, співробітник вводить посилання на вибране авто в поле «Вибраний автомобіль» і зберігає запис. Після заповнення цього поля обраний автомобіль дезактивується (при підборі автомобілів для інших клієнтів цей автомобіль не враховуватиметься) і зникає з таблиці рекомендацій. Вибір автомобіля та оновлення таблиці рекомендацій показано на рис. 4.265.

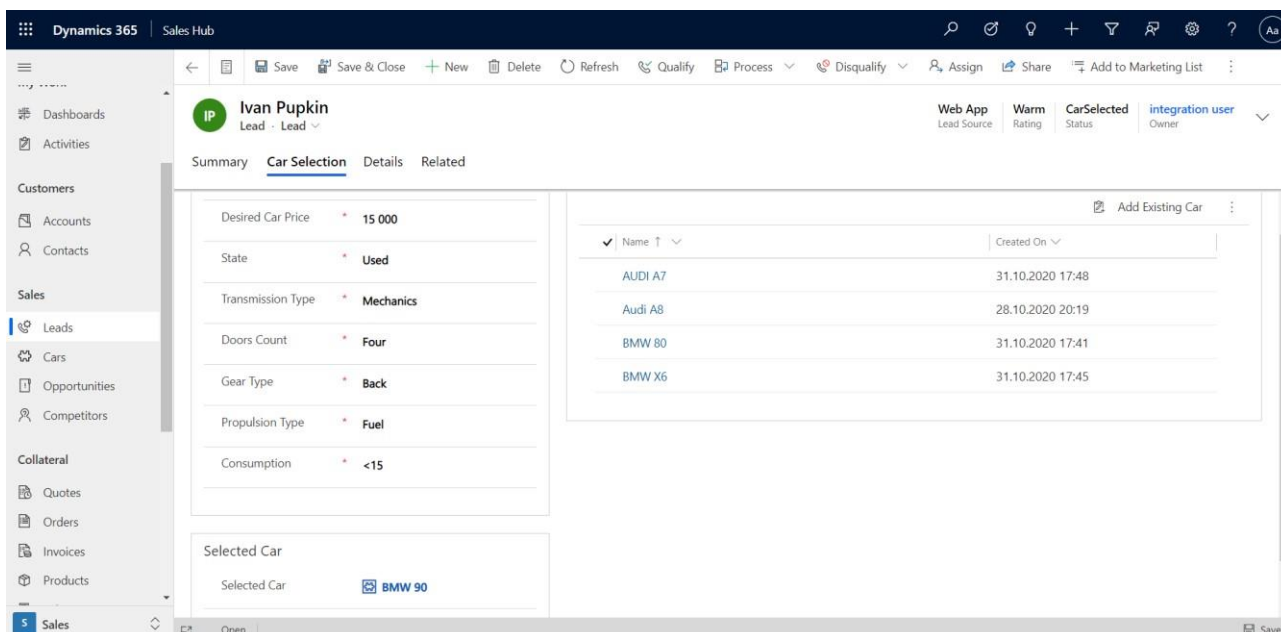


Рис. 4.26. Вибір автомобіля та оновлення таблиці рекомендацій

Якщо клієнт вибирає інший автомобіль або на даний момент передумав з вибором автомобіля, то співробітник компанії з продажу автомобілів повинен очистити поле «Вибраний автомобіль» і зберегти введене. Раніше вибраний і деактивований автомобіль буде активований, з'явиться в таблиці рекомендацій і стане доступним для пошуку рекомендацій для інших клієнтів. Ефект очищення вибраного поля автомобіля показано на рис. 4.27.

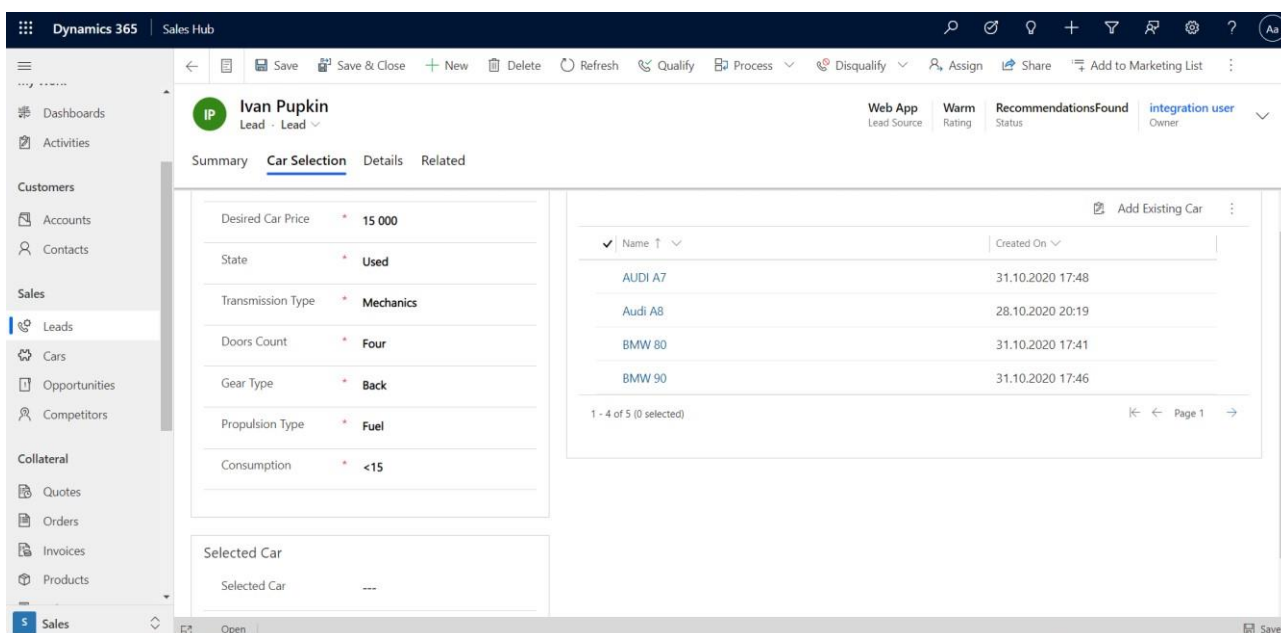


Рис. 4.27. Результат очищення поля вибраного автомобіля

Якщо клієнт відмовляється від покупки вибраного автомобіля, співробітники не зможуть зв'язатися з клієнтом за вказаними контактними даними, тоді співробітник повинен змінити статус клієнта на відповідний, натиснувши кнопку «Disqualify». Якщо статус змінюється, обраний клієнтом автомобіль стає активним і враховується при пошуку рекомендацій для інших клієнтів. Процес зміни статусу клієнта зображено на рис. 4.28.

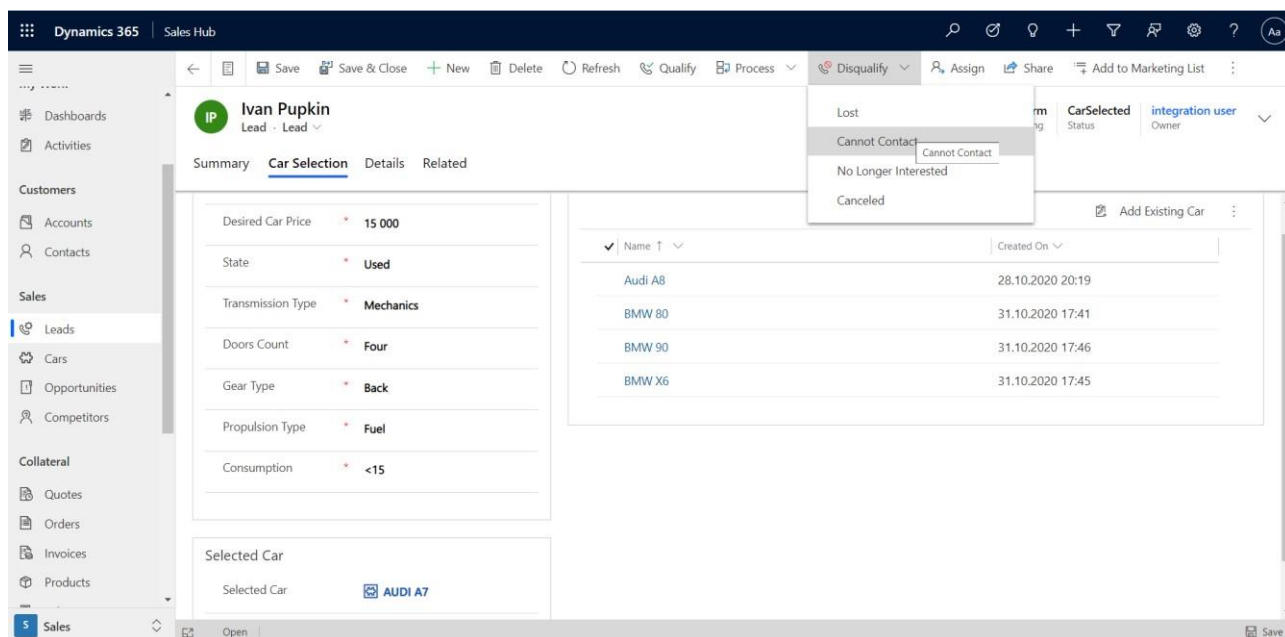


Рис. 4.28. Процес зміни статусу клієнта

Якщо клієнт придбав автомобіль, працівник повинен змінити статус клієнта на «CarBough», натиснувши кнопку «Qualify». Вибраний автомобіль стане неактивним і не використовуватиметься в майбутньому виборі автомобіля.

Результат зміни статусу клієнта на «CarBough» зображено на рис. 4.29.

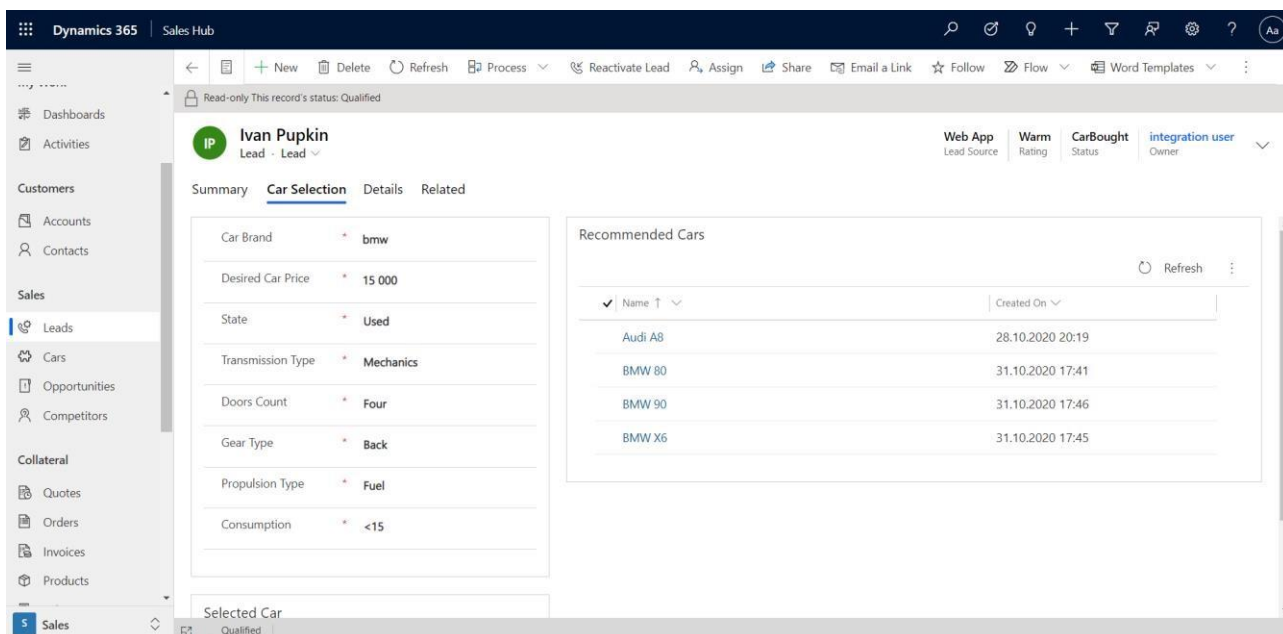


Рис. 4.29. Процес зміни статусу клієнта

Розглянемо процес ведення товарного запасу автомобіля. Щоб відобразити список доступних автомобілів в системі, натисніть кнопку “Cars” в меню зліва. Перехід до списку автомобілів і відображення списку автомобілів показано на рис. 4.30 та 4.31 відповідно.

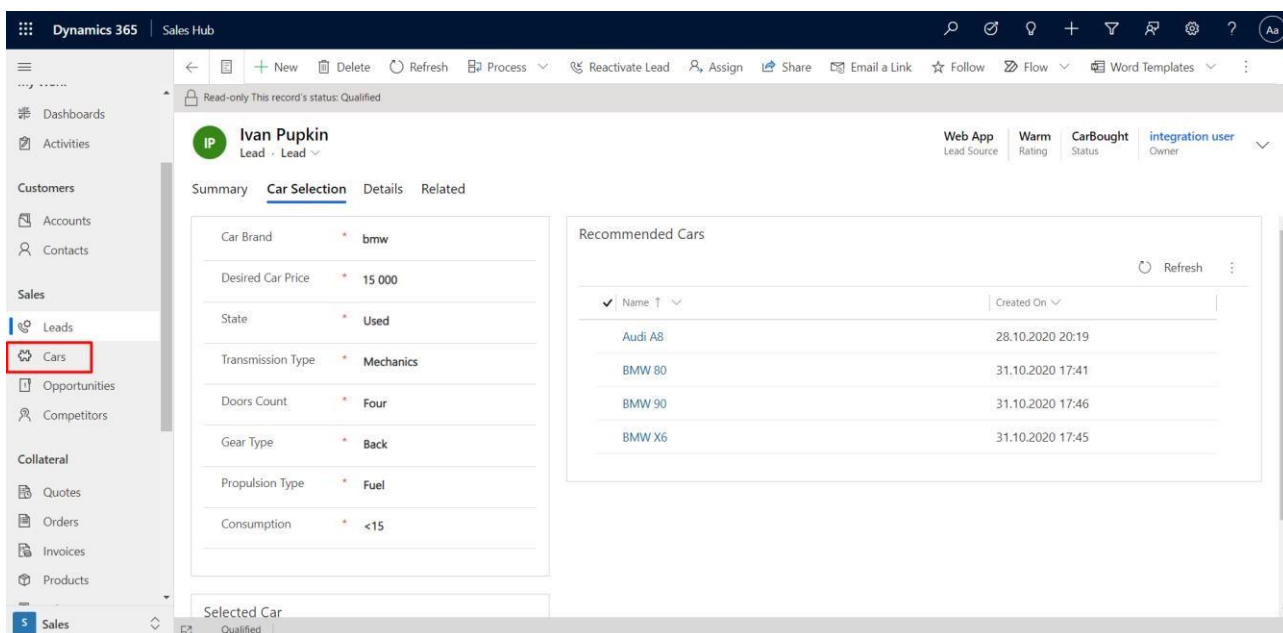
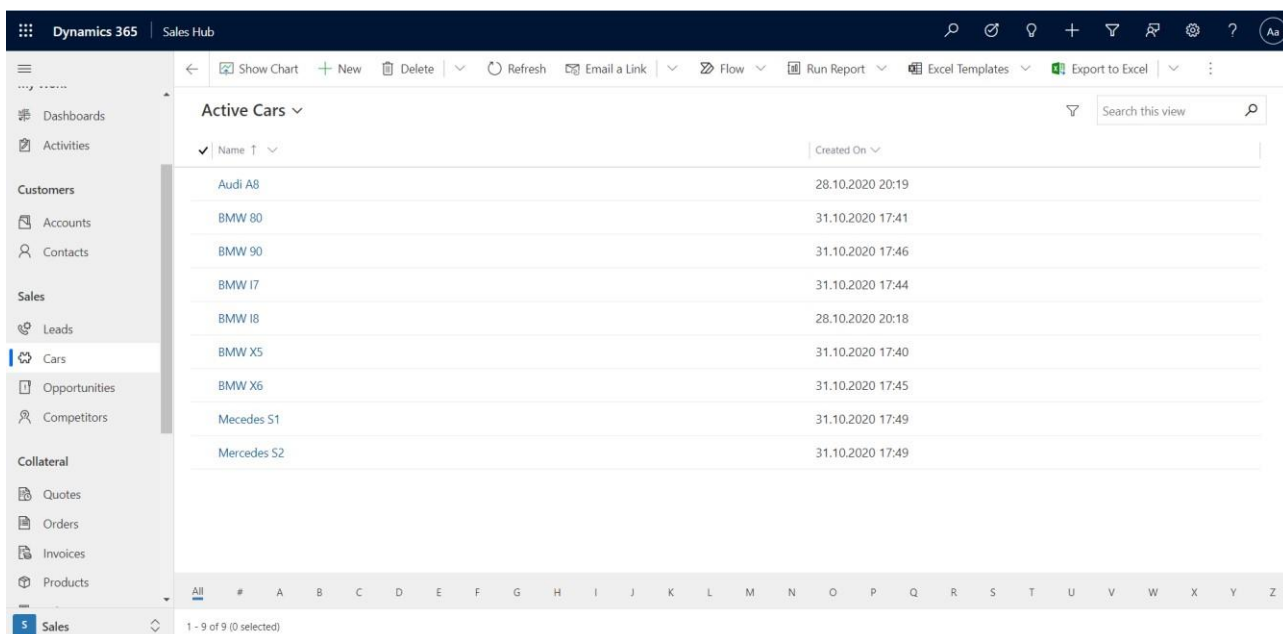


Рис. 4.30. Перехід до списку автомобілів

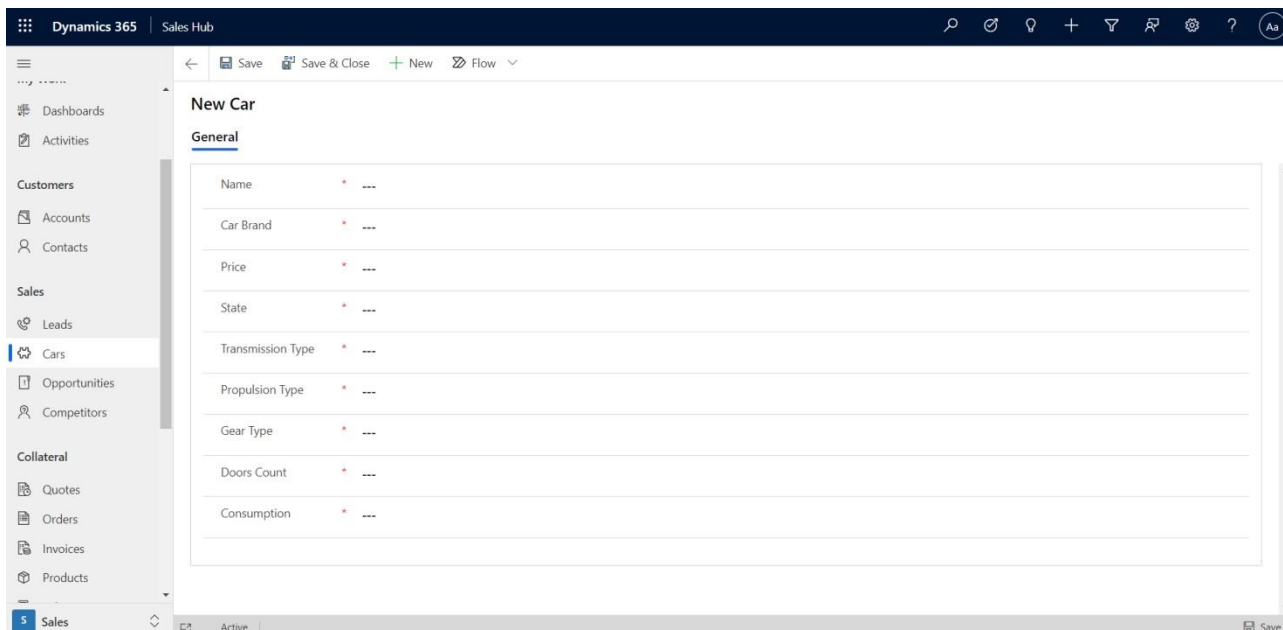


The screenshot shows the Dynamics 365 Sales Hub interface. The left sidebar contains navigation options: Dashboards, Activities, Customers (Accounts, Contacts), Sales (Leads, Cars, Opportunities, Competitors), and Collateral (Quotes, Orders, Invoices, Products). The main area displays a table titled "Active Cars" with columns for Name and Created On. The table contains 9 rows of data.

Name	Created On
Audi A8	28.10.2020 20:19
BMW 80	31.10.2020 17:41
BMW 90	31.10.2020 17:46
BMW I7	31.10.2020 17:44
BMW I8	28.10.2020 20:18
BMW X5	31.10.2020 17:40
BMW X6	31.10.2020 17:45
Mercedes S1	31.10.2020 17:49
Mercedes S2	31.10.2020 17:49

Рис. 4.31. Відображення списку автомобілів

Для того, щоб створити автомобіль в системі робітник натискає на кнопку “New” з верхнього меню. Форма створення нового автомобіля зображена на рис. 4.32.



The screenshot shows the Dynamics 365 Sales Hub interface with the "New Car" form open. The form is titled "New Car" and has a "General" tab selected. The form contains several input fields, each with a red asterisk indicating a required field:

- Name
- Car Brand
- Price
- State
- Transmission Type
- Propulsion Type
- Gear Type
- Doors Count
- Consumption

Рис. 4.32. Форма створення нового автомобіля

Робітник компанії з продажу автомобілів заповнює параметри автомобіля та натискає на кнопку “Save”. Приклад заповнення форми створення автомобіля зображено на рис. 4.33.

New Car

General

Name	* Audi A5
Car Brand	* audi
Price	* 13 300
State	* Used
Transmission Type	* Mechanics
Propulsion Type	* Fuel
Gear Type	* Full
Doors Count	* Five
Consumption	* < 10

Рис. 4.33. Приклад заповнення форми створення автомобіля

Результат створення автомобіля зображено на рис. 4.34.

Audi A5
Car

General Related

Name	* Audi A5
Car Brand	* audi
Price	* 13 300
State	* Used
Transmission Type	* Mechanics
Propulsion Type	* Fuel
Gear Type	* Full
Doors Count	* Five
Consumption	* < 10

Рис. 4.34. Форма створення нового автомобіля

Для оновлення даних запису необхідно змінити значення полів та натиснути на кнопку “Save”. Для видалення запису необхідно натиснути на кнопку “Delete”.

4.4. Опис технічного забезпечення

При створенні веб-додатку для реєстрації заявок клієнтів використовувалася архітектура клієнт-сервер, а сама CRM-система побудована на архітектурі клієнт-сервер. На схемі розгортання (рис. 4.35) показано компоненти, які використовуються в системі, і спосіб обміну даними один з одним.

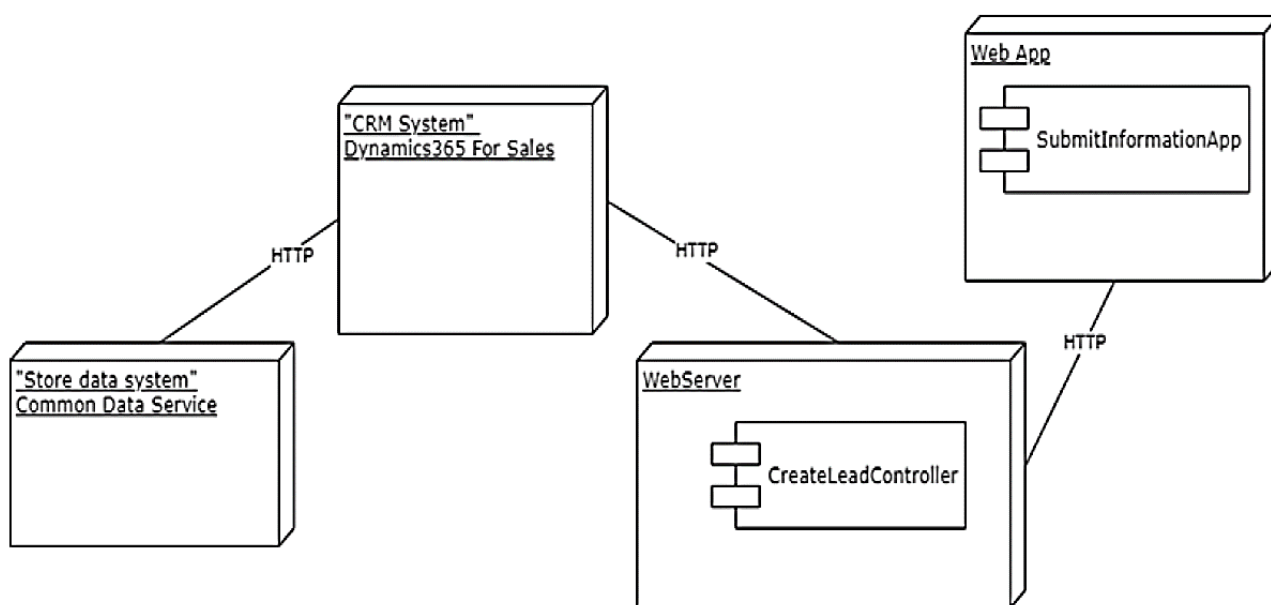


Рис. 4.35. Структурна схема розгортання

Перевагою розробленої архітектури є те, що замість веб-додатку можна використовувати будь-яке джерело даних, наприклад, чат-бот чи іншу систему. Ви також можете надсилати заявки з кількох джерел до системи рекомендацій одночасно.

Висновки до розділу 4

У частині, що описує програмно-технічне забезпечення, описуються технології та інструменти, які були використані для створення системи рекомендацій для вибору автомобілів для продажу клієнтам, розроблені блок-схеми класів, блок-схема послідовності, блок-схема впровадження, а також розроблено посібники користувача.

РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

5.1 Опис ідеї проєкту

В описі ідеї проєкту описується зміст проєкту, напрями застосування проєкту та вигоди для користувача. Опис ідеї проєкту представлено у табл. 5.1.

Таблиця 5.1

Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
1	2	3
<p>Рекомендаційна система підбору автомобілів для продажу клієнтам. Для заповнення заявок на підбір автомобілів було розроблено веб застосунок в якому клієнт заповнює власні дані та побажання щодо автомобіля. Для управління процесом роботи з клієнтом та знаходження рекомендацій було розгорнуто та модифіковано CRM систему.</p>	<p>Використання рекомендаційної системи для знаходження рекомендацій для клієнтів після отримання звернень.</p>	<ul style="list-style-type: none"> - зменшення витрат часу робітників компанії на знаходження рекомендацій; - збільшення кількості оброблених звернень за певний період часу.
	<p>Використання рекомендаційної системи для управління процесом по роботі з клієнтом.</p>	<ul style="list-style-type: none"> - збільшення точності відслідковування статусу обробки заявки від клієнта; - забезпечення існування єдиної бази звернень;

1	2	3
	Використання рекомендаційної системи для управління даними клієнтів та автомобілів.	Забезпечення єдиного місця збереження даних про клієнтів та автомобілі, що допомагає швидко знаходити необхідні дані та мінімізує втрату даних.

Щоб зрозуміти, чи є проєкт конкурентоспроможним, виявляються слабкі, нейтральні та сильні сторони, порівняно з проєктами конкурентів. Визначення сильних, слабких та нейтральних характеристик проєктної ідеї наведено в табл. 5.2.

Таблиця 5.2

Дослідження сильних, слабких та нейтральних характеристик ідей проєкту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні)товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	carandbike	padaguides	infocar			
1	2	3	4	5	6	7	8	9
1	Легкість реалізації	Легко реалізувати	Вже реалізований	Вже реалізований	Вже реалізований		+	

1	2	3	4	5	6	7	8	9
2	Знаходження рекомендацій	Рекомендаційна система знаходить рекомендації, а не повертає всі автомобілі профільтовані по обраним клієнтом параметрами.	Знаходять рекомендації	Повертає всі автомобілі профільтовані по обраним клієнтом параметрами.	Повертає всі автомобілі профільтовані по обраним клієнтом параметрами.			+
3	Збереження вхідних заявок від клієнтів	Рекомендаційна система зберігає заявки	Дані не знайдено	Дані не знайдено	Дані не знайдено			+
4	Моментальне відображення рекомендацій	Відсутнє	Наявне	Наявне	Наявне	+		

1	2	3	4	5	6	7	8	9
5	Зручний інтерфейс по управлінню даними клієнтів, автомобілів	Інтерфейс - зручний та зрозумілий	Дані не знайдено	Дані не знайдено	Дані не знайдено			+
6	Легкість інтеграції рекомендаційної системи з іншими джерелами заявок на підбір автомобіля	Легко інтегрується з іншими системами за рахунок існування Web API	Заявки отримуються лише в веб застосунку	Заявки отримуються лише в веб застосунку	Заявки отримуються лише в веб застосунку			+
7	Швидкість знаходження рекомендацій	Висока	Висока	Висока	Висока		+	
8	Можливість отримувати заявки на знаходження рекомендацій з декількох джерел	Наявна	Відсутня	Відсутня	Відсутня		+	

1	2	3	4	5	6	7	8	9
9	Можливість розмістити серверну частину веб застосунка, як на серверах, так і в хмарі	Наявна	Дані не знайдено	Дані не знайдено	Дані не знайдено			+
10	Легкість для подальшої модифікації системи по роботі з даними автомобілів, клієнтів та управління процесом по роботі з клієнтом	Легко додається нова бізнес логіка	Дані не знайдено	Дані не знайдено	Дані не знайдено			+
11	Масштабованість системи	При збільшенні кількості наявних автомобілів в системі процес пошуку рекомендацій відбувається швидко	Дані не знайдено	Дані не знайдено	Дані не знайдено			+

1	2	3	4	5	6	7	8	9
12	Можливість використання даних клієнтів в маркетингових цілях	Легко можна інтегрувати маркетинговий	Відсутня	Відсутня	Відсутня			+

5.2. Розроблення ринкової стратегії

Розробимо стратегію охоплення ринку та базову стратегію розвитку. Вибір цільових груп потенційних споживачів представлено табл. 5.3 визначення базової стратегії розвитку – в табл. 5.4.

Таблиця 5.3

Цільові групи потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Автомобільні салони	Висока	Високий	Низька	Середня
2	Компанії з продажу автомобілів	Висока	Високий	Низька	Середня
3	Особи підприємці, які продають автомобілі	Середня	Середній	Низька	Середня

Було обрано три цільові групи, оскільки потреби кожної з них не відрізняються одна від одної. Різниця між ними лише в обсязі продажів та обсязі доступних даних. Вибравши три цільові групи, досягається більше охоплення ринку та більш швидке зростання репутації.

Тому будемо використовуватимемо стратегію масового маркетингу.

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розгортання системи на середовищі розробки, тестування системи, відлагодження системи, тестування	Стратегія масового маркетингу	Стабільність пропонованої системи, відсутність помилок, що будуть заважати роботі системи.	Стратегія спеціалізації
системи фокус групами, фінальне налаштування системи, виведення на ринок.			

Дослідження базової стратегії конкурентної поведінки в табл. 5.5.

Дослідження базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першорхідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	Не повністю	Передбачається впровадження систем у нових споживачів і зі збільшенням репутації планується брати участь в тендерах вже існуючих користувачів	Ні	Стратегія виклику лідера

Дослідження стратегії позиціонування наведено в табл. 5.6.

Таблиця 5.6

Дослідження стратегії позиціонування

п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Виконана з забезпеченням безпеки даних в системі	Стратегія спеціалізації	Стабільність пропонованої системи, відсутність помилок, що будуть заважати роботі системи.	Надійність збереження даних
2	Підтримка проекту та подальша співпраця для розширення функціонала системи за необхідністю.	Стратегія спеціалізації		Високий рівень підтримки після впровадження
3	Стабільність роботи системи.	Стратегія спеціалізації		Готовність модернізувати систему під потреби клієнта Стабільна робота системи

5.3. Розроблення маркетингової програми

За підсумками дослідження розробимо концепцію маркетингу продукції.

Дослідження ключових переваг концепції потенційного продукту наведено у табл. 5.7.

Дослідження ключових переваг концепції потенційного товару

п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Потреба у рекомендаційній системі, яка зменшить час знаходження рекомендацій і тим самим збільшить обсяги оброблених заявок.	зменшення витрат часу робітників компанії на знаходження рекомендацій;	автоматичне знаходження рекомендацій
		збільшення точності відслідковування статусу обробки заявки від клієнта	використання CRM системи, як системи для управління процесом по роботі з клієнтом
		забезпечення єдиного місця збереження даних про клієнтів та автомобілі, що допомагає швидко знаходити необхідні дані та мінімізує втрату даних.	використання CRM системи, як системи для збереження та ведення даних

Опис трьох рівнів моделей товару наведений в табл. 5.8.

Опис трьох рівнів моделей товару

Рівні товару	Сутність та складові		
1	2		
I. Товар за задумом	Рекомендаційна система підбору автомобілів для продажу клієнтам. Система складається з CRM системи з інтегрованим алгоритмом для знаходження рекомендацій та веб-застосунку для заповнення заявок.		
II. Товар у реальному виконанні	Властивості/характеристики		Вр/Тх /Нм /Тл/Е/Ор

Рівні товару	Сутність та складові		
1	2		
II. Товар у реальному виконанні	Властивості/характеристики		Вр/Тх /Нм /Тл/Е/Ор
	Збереження вхідних заявок від клієнтів Легкість інтеграції рекомендаційної системи з іншими джерелами заявок на підбір автомобіля Можливість отримувати заявки на знаходження рекомендацій з декількох джерел Масштабованість системи Можливість розмістити серверну частину веб застосунка, як на серверах, так і в хмарі		
	Рекомендаційна система відтестована, працює стабільно і задовольняє вимогам для виходу на ринок.		
	Кастомізація CRM системи з інтегрованим рекомендаційним алгоритмом буде поставлятися, як промаркований незмінний пакет, веб-застосунок буде поставлятися у вигляді архіву необхідних бібліотек і компонентів.		
	Марка: ASCarRec		

Система буде захищена від копіювання шляхом патентування алгоритму та принципу роботи системи.

Потім сформуємо систему продажу та концепцію маркетингових комунікацій. Створення системи продаж представлено у табл. 5.9. Поняття маркетингової комунікації наведено у табл. 5.10.

Дослідження системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибини каналу збуту	Оптимальна система збуту
1	Підписання контракту на впровадження і підтримку системи	Впровадження системи на замовника середовищі	Однорівнева	Власна система збуту

Таблиця 5.10

Дослідження концепції маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Перехід за рекламним посиланням	Електронна пошта, контекстна реклама в інтернеті	Рекомендаційна система для знаходження рекомендацій на основі побажань клієнта	Ознайомити потенційних клієнтів з можливостями і рекомендаційної системи та зацікавити до впровадження	Розробка маркетингових повідомлень, описів для маркетингових посилань, розробка сайту компанії з описом функціоналу рекомендаційної системи

5.4. Вимоги до технічного та програмного забезпечення

Проведемо технологічний аудит технологій, необхідних реалізації проєкту. Перелік технологій, які будуть використовуватися при реалізації

проєкту, аналіз наявності та доступності необхідних технологій наведено у табл. 5.11.

Таблиця 5.11

Технологічне рішення ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Знаходження рекомендацій на основі побажань клієнта	Алгоритм знаходження рекомендацій на основі побажань клієнта	Наявні	Доступні
2	Зберігання та ведення даних клієнтів, звернень, автомобілів в системі	Будь-яка CRM система	Наявні	Доступні
3	Управління процесом по роботі з клієнтом	Будь-яка CRM система	Наявні	Доступні
4	Отримання заявок на знаходження рекомендованих автомобілів	Веб застосунок розроблений за допомогою будь-яких технологій	Наявні	Доступні
5	Збереження заявок на знаходження рекомендованих автомобілів	Web API розроблені за допомогою .NetCore	Наявні	Доступні

Технологічна реалізація проєкту можлива. Для пошуку рекомендацій було обрано алгоритм пошуку рекомендацій, з побажань замовника. Для зберігання та ведення даних про автомобілі, клієнтів, запити та управління процесом роботи з клієнтами було обрано CRM-систему Microsoft Dynamics 365 for Sales. Для розробки веб-API було обрано платформу .NetCore 2.2, яка використовує мову С#. Для створення веб-програми для заповнення заявок на пошук рекомендованих автомобілів було обрано фреймворк Bootstrap та бібліотеку JQuery на основі мови програмування JavaScript.

Висновки до розділу 5

У частині, що стосується розробки стартап-проекту, було зроблено опис ідеї, проведено технологічний аудит ідеї проекту, проаналізовано ринкові можливості для запуску стартап-проекту, визначено фактори конкурентоспроможності та визначено ринкову та маркетингову стратегію.

У результаті аналізу можна дійти до висновку, що рекомендаційна система має бути затребувана ринком, оскільки немає систем з аналогічним функціоналом. Було обрано наступний варіант поведінки: впровадити систему у середовище розробки, протестувати систему, налагодити систему, протестувати систему через фокус-групи, зробити остаточне налаштування системи, запустити її на ринку. Така альтернатива гарантує, що при виході системи на ринок, вона працюватиме стабільно і без системних помилок. Таким чином, слід зробити висновок про доцільність подальшого розвитку цієї системи.

ВИСНОВКИ

У результаті роботи над магістерською роботою було досліджено існуючий бізнес-процес пошуку рекомендацій щодо бажаних характеристик для клієнта компанії, встановлено, що пошук здійснює сам керівник, а після аналізу доступних методів і алгоритми пошуку рекомендацій виявили, що вони не дозволяють знайти рекомендації, якщо у клієнта немає історії в компанії. Для клієнтів, які не мають історії в компанії, була розроблена концептуальна модель процесу автоматичного пошуку рекомендацій за методологією IDEF0.

Створено алгоритм пошуку рекомендацій для клієнтів, які не мають історії в компанії. Алгоритм заснований на методах контент-фільтрації та пошуку лексикографічної відповідності між вектором бажаних ознак і векторами характеристик автомобіля, доступних компанії.

Експериментально перевірено, що час роботи алгоритму залежить від збільшення кількості векторів характеристик автомобіля на підприємстві, а при збільшенні кількості в 25 разів час збільшується в 5 разів, що для 2500 тис. характерних векторів досягає менше однієї секунди. Крім того, середнє значення коефіцієнта узгодження зменшується зі збільшенням загальної кількості векторів.

Експерименти показали, що алгоритм стійкий до збільшення загальної кількості векторів характеристик автомобіля в системі.

В результаті дослідження розроблено архітектуру системи рекомендацій, що складається з клієнтської та серверної частин веб-додатку, CRM-системи на платформі Dynamics 365, сервісу CDS та переваг розробленої архітектури.

На основі розробленої архітектури реалізована система рекомендацій на платформі Dynamics 365 з використанням розробленого алгоритму.

За результатами розробки стартап-проекту можна зробити висновок, що система рекомендацій повинна мати попит на ринку, оскільки систем з подібним функціоналом на ринку немає. Обрана альтернатива ринкової поведінки при

виході системи на ринок забезпечить стабільність її функціонування. Тому слід зробити висновок про доцільність подальшого розвитку системи.

Результатом магістерської роботи є дослідження, яке призвело до створення алгоритму пошуку рекомендацій з продажу автомобіля клієнтам, які не мають історії в компанії. Розроблена система рекомендацій реалізує створений алгоритм, який дозволить досягти мети дослідження – підвищення ефективності роботи менеджерів, що в цілому збільшить прибуток компанії за рахунок збільшення загальної кількості оброблених заявок і продажів.

СПИСОК ЛІТЕРАТУРИ

1. Арцибасов В. Є. Методи та засоби побудови рекомендаційних систем для задач електронної комерції : автореферат дисертації на здобуття наукового ступеня кандидата технічних наук : 05.13.06 – інформаційні технології / Віталій Євгенович Арцибасов ; Міністерство освіти і науки України, Національний університет «Львівська політехніка». – Львів, 2015. – 28 с. – Бібліографія: с. 19–21 (24 назви).
2. Горборуков В. В. Створення рекомендаційної системи підтримки прийняття рішень для запису на вибіркові навчальні дисципліни / В. В. Горборуков, О. В. Олецкий // Наукові записки НаУКМА. Комп'ютерні науки. - 2017. - Т. 198. - С. 54-58. - Режим доступу: http://nbuv.gov.ua/UJRN/NaUKMAkn_2017_198_12.
3. Жежерун О. П. Побудова рекомендаційних систем на основі онтологій / Жежерун О. П., Яремко С. А. // Наукові записки НаУКМА. Комп'ютерні науки. -2017. - Т. 198. - с. 36-41, Електронний ресурс: <http://ekmair.ukma.edu.ua/handle/123456789/12536>
4. Ковалевський Л. Г. Світовий автомобільний ринок: сучасний стан, особливості та перспективи розвитку / Л. Г. Ковалевський, Н. Ю.
5. Коровайченко // Зовнішня торгівля: економіка, фінанси, право. - 2015. - № 5-6. - С. 60–67. - Режим доступу: http://nbuv.gov.ua/UJRN/uazt_2015_5-6_8.
6. Кучерук В. Ю. Покращення алгоритму "itemtoitem" методу колаборативної фільтрації для розробки рекомендаційних систем на основі косинусної міри шляхом оцінки релевантності / В. Ю. Кучерук, М. В. Глушко // ScienceRise. - 2018. - № 1. - С. 20-24. - Режим доступу: http://nbuv.gov.ua/UJRN/texc_2018_1_6.
7. Лобур М. Побудова асоціативних правил для прогнозування рекомендацій в колаборативних рекомендаційних системах / М.Лобур, Ю.Стех, М.Шварц // Збірник наукових праць Української Академії Друкарства. Квалілогія книги. Львів. – 2017. – № 2 (32). – С. 82–86

8. Лобур М., Стех Ю., Шварц М. Метод і алгоритм прогнозування рекомендацій для спільнот користувачів / М.Лобур, Ю.Стех., М.Шварц. // Збірник наукових праць Української Академії Друкарства. Квалілогія книги. Львів, 2017. – № 1 (31). – С. 88–93
9. Мелешко Є. В. Дослідження робастності рекомендаційних систем з колаборативною фільтрацією до інформаційних атак / Є. В. Мелешко, В. Д. Хох, О. С. Улічев // Кібербезпека: освіта, наука, техніка. - 2019. - № 1. - С. 95104. - Режим доступу: http://nbuv.gov.ua/UJRN/cest_2019_1_11.
10. Негрей М. В. Компаративний аналіз ефективності рекомендаційних систем в маркетингу / М. В. Негрей, Т. В. Гнот // Вісник Хмельницького національного університету. Економічні науки. - 2017. - № 5. - С. 278-286. - Режим доступу: http://nbuv.gov.ua/UJRN/Vchnu_ekon_2017_5_53.
11. Офіційний сайт мови JavaScript – [Електронний ресурс] // Режим доступу: <https://www.javascript.com/>
12. Статистика автопродаж [Електронний ресурс] – Режим доступу до ресурсу: <https://proautomoto.com/category/202-2020>.
13. Стоволос Н. Б. Дослідження алгоритмів рекомендаційних сервісів з персоналізованим ціноутворенням [Електронний ресурс] / Н. Б. Стоволос, В. В. Падалка // Збірник наукових статей молодих учених, аспірантів та студентів СНАУ. – Суми : СНАУ, 2018. - С. 248-252.
14. Шингалов Д. В. Математична модель рекомендаційної системи з врахуванням емоційного забарвлення коментарів у якості контексту / Д. В. Шингалов, Є. В. Мелешко, Р. М. Минайленко, В. А. Резніченко // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. - 2018. - Вип. 31. - С. 181-186. - Режим доступу: http://nbuv.gov.ua/UJRN/znpkntu_2018_31_24.
15. Щербак Д. В. Система рекомендації навчальних матеріалів / Д. В. Щербак, О. П. Сирота // Вчені записки Таврійського національного університету імені В.

- I. Вернадського. Серія : Технічні науки. - 2018. - Т. 29(68), № 6(2). - С. 26-29.
- Режим доступу: [http://nbuv.gov.ua/UJRN/sntuts_2018_29\(68\)_6\(2\)__8](http://nbuv.gov.ua/UJRN/sntuts_2018_29(68)_6(2)__8).
16. Щербань В. С., Гайдейчук Ю. А. Рекомендаційна система вибору відеофільмів; Google Scholar. – 2016. – 4 с., Електронний ресурс: <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10841/573.pdf?sequence=3>.
17. C# programming guide– [Електронний ресурс] // Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
18. Chalyi S. Ситуаційна модель користувачького вибору в рекомендаційній системі / S. Chalyi, I. Pribylnova // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 2 (54). – С. 159-163. – doi:<https://doi.org/10.26906/SUNZ.2019.2.159>.
19. Dyrix – [Електронний ресурс] // Режим доступу: <https://github.com/AlexanderGurinov/Dyrix>
20. IOrganizationServiceInterface, Електронний ресурс:<https://docs.microsoft.com/ru-u/dotnet/api/microsoft.xrm.sdk.iorganizationservice?view=dynamics-general-ce-9>
21. Isinkaye F. O. Recommendationsystems: Principles, methodsandevaluation / F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh. // Egyptian Informatics Journal. – 2015. – №16. – С. 261–273.
22. Meleshko, Y. (2018) Проблеми сучасних рекомендаційних систем та методи їх рішення, Системи управління, навігації та зв'язку. Збірник наукових праць. Полтава: ПНТУ, 4(50), с. 120-124. doi: <https://doi.org/10.26906/SUNZ.2018.4.120>.
23. Meleshko, Ye. (2018), "Quality assessment methods of work of recommendation systems", Academic Journal "Control, Navigation and Communication Systems", Poltava National Technical Yuri Kondratyuk University, No. 5 (51), P. 92–97, DOI: <https://doi.org/10.26906/SUNZ.2018.5.092> (in Ukrainian)
24. MeleshkoE.V. Дослідження методів побудови рекомендаційних систем в мережі інтернет / MeleshkoE.V., S.G. Semenov, V.D. Khokh // Системи

- управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 1 (47). – С. 131-136. – doi:<https://doi.org/10.26906/SUNZ.2018.1.131>.
25. Nuget – [Електронний ресурс] // Режим доступу: <https://www.nuget.org/>
26. Perspectives of world science and education. Abstracts of the 9th International scientific and practical conference. CPN Publishing Group. Osaka, Japan. 2020. Pp. 886-893. URL: <https://sci-conf.com.ua>.
27. Philippov, S. A., Zakharov, V. N., Stupnikov, S. A., and Kovalev, D. Y.
28. What is Common Data Service, Електронний ресурс:
<https://docs.microsoft.com/en-us/powerapps/maker/common-data>
<https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/data-platform-introservice/data-platform-intro>.

