

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Механіко – технологічний факультет

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

В.о. завідувача кафедри

технічного сервісу та інженерного

(назва кафедри)

менеджменту імені М.П. Момотенка

Руслан ШАТРОВ

(підпис)

(ПБ)

«___» _____ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Удосконалення інженерного менеджменту системи підбору продажу клієнтам ТОВ
«Віді-Автострада»»

Спеціальність 274 «Автомобільний транспорт»

(код і назва)

Освітня програма «Автомобільний транспорт»

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна, або освітньо-наукова)

Гарант освітньої програми

доктор технічних наук, професор

(науковий ступінь та вчене звання)

(підпис)

Войтюк Валерій Дмитрович

(ПБ)

Керівник магістерської кваліфікаційної роботи

д.т.н., професор

(науковий ступінь та вчене звання)

(підпис)

Роговський Іван Леонідович

(ПБ)

Виконав

(підпис)

Каталінчук Богдан Ігорович

(ПБ)

КИЇВ – 2025

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Механіко – технологічний факультет

ЗАТВЕРДЖУЮ

Завідувач кафедри технічного сервісу та інженерного менеджменту імені М.П. Момотенка

д.т.н., проф. Іван РОГОВСЬКИЙ
(науковий ступінь, вчене звання) (підпис) (ПІБ)

« ____ » _____ 2024 р.

**З А В Д А Н Н Я
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ**

Каталінчуку Богдану Ігоровичу

(прізвище, ім'я, по батькові)

Спеціальність 274 «Автомобільний транспорт»

(код і назва)

Освітня програма «Автомобільний транспорт»

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна, або освітньо-наукова)

Тема магістерської кваліфікаційної роботи «Удосконалення інженерного менеджменту системи підбору продажу клієнтам ТОВ «Віді-Автострада»»

затверджена наказом ректора НУБіП України від «13» листопада 2024 р. № 2039 «С»

Термін подання завершеної роботи на кафедру _____

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи Науково – технічна література; результати науково-дослідних робіт по літературних джерелах по вивченню питання інженерного менеджменту системи підбору продажу клієнтам

Перелік питань, що підлягають дослідженню:

- 1) РОЗДІЛ 1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ПІДБОРУ АВТОМОБІЛІВ ДЛЯ ПРОДАЖУ КЛІЄНТАМ
- 2) РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ ПОШУКУ РЕКОМЕНДАЦІЙ
- 3) РОЗДІЛ 3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ
- 4) РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Перелік графічного матеріалу Електронна презентація на 16 слайдах

Дата видачі завдання «11» листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____

(підпис)

Роговський І.Л.

(прізвище та ініціали)

Завдання прийняв до виконання _____

(підпис)

Каталінчук Б.І.

(прізвище та ініціали)

РЕФЕРАТ

Магістерська кваліфікаційна робота: 83 с., 38 рис., 32 табл., 36 джерел.

Актуальність. Сучасний глобальний ринок автомобілів є однією із важливих складових економіки світу, одним з найконкурентніших ринків. Продаж автомобілів напряму позначається на економічному зростанні та кризових явищах глобальної економіки [1].

Світовий авторинок у вересні поточного року підвищився на 0,5% до 7647000 легкових автомобілів. Найбільшу кількість автомобілів за звітний період реалізували в КНР з показником в 2,487 млн од., що на 9,3% вище за минулорічний результат. Американські автолюбители придбали 1,337 млн автомобілів, що на 4,4% більше, ніж в 2019 році. Західноєвропейські авторинки продемонстрували результат в 1,367 млн автомобілів (+1,3%). Реалізація автомобілів в Східній Європі підвищилася на 19,7% до 393,3 тис. шт. [2].

Через високий попит на даний товар автосалони, компанії, що займаються продажами автомобілів отримують велику кількість звернень від клієнтів на покупку автомобілів. Відповідно робітники відділу продажу отримують та аналізують побажання кожного клієнта та, зазвичай, власноруч підбирають найкращі варіанти для клієнтів. Проте такий процес підбору займає досить багато часу і залежить від кваліфікації робітника, що знаходить рекомендації, тобто знайдені рекомендації залежать від людського фактору і не завжди можуть бути правильними. Тому обраний напрям дослідження і пошуку рекомендаційної системи є актуальним.

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

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

- проаналізувати існуючі методи та засоби і визначити проблеми при пошуку рекомендацій;
- створити та дослідити алгоритм для пошуку рекомендацій;

➤ розробити архітектуру рекомендаційної системи на платформі Dynamics 365;

➤ реалізувати рекомендаційну систему на платформі Dynamics 365.

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

Предмет дослідження - методи та алгоритми пошуку рекомендацій.

Наукова новизна одержаних результатів полягає у знаходженні рекомендацій на основі побажань клієнта без попередньо існуючої історії для певного клієнта.

Ключові слова: РЕКОМЕНДАЦІЙНА СИСТЕМА, ЛЕКСИКОГРАФІЧНА ПОДІБНІСТЬ, ЕВКЛІДОВА ВІДСТАНЬ, АВТОМОБІЛЬ, CRM СИСТЕМА, ІНТЕГРАЦІЯ.

ЗМІСТ

РЕФЕРАТ	3
ЗМІСТ	5
ВСТУП	7
РОЗДІЛ 1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ПІДБОРУ АВТОМОБІЛІВ ДЛЯ ПРОДАЖУ КЛІЄНТАМ	8
1.1. Опис бізнес - процесів	8
1.1.1. Опис процесу діяльності	8
1.1.2. Актори і функції	9
1.1.3. Структурна схема діяльності з підбору рекомендацій	10
1.2. Рішення з інформаційного забезпечення	11
1.3. Висновки до розділу 1	13
РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ ПОШУКУ РЕКОМЕНДАЦІЙ	14
2.1. Змістовна постановка задачі	14
2.2. Математична постановка задачі	14
2.3. Огляд методів розв'язання	14
2.4. Розробка алгоритму пошуку рекомендацій	26
2.5. Приклад роботи алгоритму	29
2.6. Результати експериментальних досліджень	35
2.7. Висновки до розділу 2	37
РОЗДІЛ 3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	39
3.1. Засоби розробки	39
3.2. Архітектура програмного забезпечення	43
3.2.1. Структурна схема класів	43
3.2.2. Структурна схема послідовності	49
3.3. Інструкція користувача	49
3.3.1. Інструкція клієнта	51

3.3.2. Інструкція робітника компанії з продажу автомобілів	60
3.4. Опис технічного забезпечення	61
3.5. Висновки до розділу 3	62
РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	62
4.1. Опис ідеї проекту	64
4.2. Технологічний аудит ідеї проекту	65
4.3. Аналіз ринкових можливостей запуску стартап-проекту	69
4.4. Розробка ринкової стратегії проекту	75
4.5. Розроблення маркетингової програми стартап-проекту	77
4.6. Висновки до розділу 4	80
ВИСНОВКИ	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	84

ВСТУП

Сучасний глобальний ринок автомобілів є однією із важливих складових економіки світу, одним з найконкурентніших ринків. Продаж автомобілів напряду позначається на економічному зростанні та кризових явищах глобальної економіки [1].

Світовий авторинок у вересні поточного року підвищився на 0,5% до 7647000 легкових автомобілів [2].

Через високий попит на даний товар автосалони, компанії, що займаються продажами автомобілів отримують велику кількість звернень від клієнтів на покупку автомобілів. Відповідно робітники відділу продажу отримують та аналізують побажання кожного клієнта та, зазвичай, власноруч підбирають найкращі варіанти для клієнтів. Проте такий процес підбору займає досить багато часу і залежить від кваліфікації робітника, що знаходить рекомендації, тобто знайдені рекомендації залежать від людського фактору і не завжди можуть бути правильними. Тому обраний напрям дослідження і створення рекомендаційної системи є актуальним.

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

Для досягнення мети необхідно виконати наступні завдання:

- проаналізувати існуючі методи та засоби і визначити проблеми пошуку рекомендацій;
- створити та дослідити алгоритм для пошуку рекомендацій;
- розробити архітектуру рекомендаційної системи на платформі Dynamics 365;
- реалізувати рекомендаційну систему на платформі Dynamics 365.

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

Предмет дослідження - методи та алгоритми пошуку рекомендацій.

РОЗДІЛ 1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ПІДБОРУ АВТОМОБІЛІВ ДЛЯ ПРОДАЖУ КЛІЄНТАМ

1.1. Опис бізнес - процесів

1.1.1. Опис процесу діяльності

Перед покупкою автомобіля клієнт заповнює на сайті компанії, що продає автомобілі, або в режимі телефонного спілкування, свої побажання по майбутньому автомобілю. Після чого робітники відділу продажу аналізують побажання кожного клієнта та власноруч підбирають найкращі варіанти для клієнтів. Після чого записують собі варіанти в будь-якому вигляді. Таким чином, через відсутність автоматизації оброблення інформації дані можуть втратитися і пошук треба проводити ще раз. Після знаходження рекомендацій працівник відділу продажів повідомляє про можливі варіанти автомобілів клієнту. Якщо клієнт після комунікації робить покупку автомобіля, то робітник відділу продажу має видалити даний автомобіль зі всіх джерел, по яким він знаходить рекомендації. Бізнес-процес опрацювання запиту клієнта на підбір автомобіля до впровадження рекомендаційної системи наведено на рисунку 1.1 за допомогою AS-IS діаграми IDEF0.

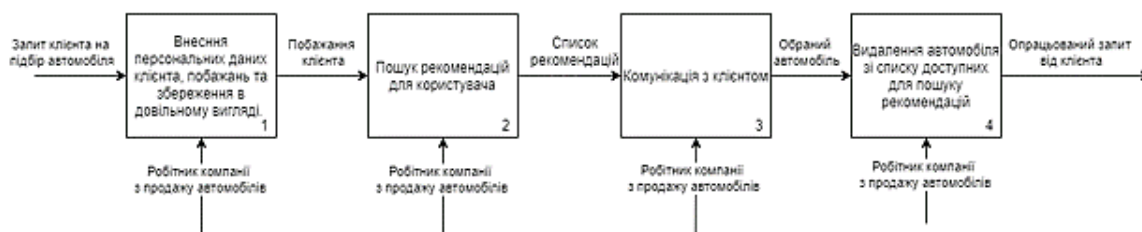


Рис. 1.1. Діаграма бізнес-процесу до впровадження рекомендаційної системи

Після впровадження рекомендаційної системи бізнес процес змінюється, і більшість задач буде виконувати система. Після запиту від клієнта персональні дані та побажання клієнта зберігаються в системі. Після збереження даних робітник компанії з продажу автомобілів починає роботу з певним клієнтом,

система після початку роботи автоматично знаходить рекомендації та змінює статус клієнта. Наступним кроком є комунікація з клієнтом для того, щоб надати знайдений список рекомендацій. При виборі клієнтом автомобіля робітник компанії з продажу автомобілів на записі користувача в CRM системі обирає автомобіль. У разі коли клієнт купує або відмовляється від покупки автомобіля робітник компанії змінює статус клієнта на «Придбав автомобіль» або «Відмовився від покупки». Якщо клієнт придбав автомобіль, то запис автомобіля в CRM системі деактивується, він не враховується в подальшому знаходженні рекомендацій. У випадках, коли клієнт відмовляється від покупки автомобіля, обраний автомобіль, що був деактивований, активується. Бізнес-процес опрацювання запиту клієнта на підбір автомобіля після впровадження рекомендаційної системи наведено на рисунку 1.2. за допомогою TO-VE діаграми IDEF0.



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

1.1.2. Актори і функції

Перед розробкою діаграми варіантів слід визначити акторів системи, після чого дії, які будуть виконувати актори. Система має двох акторів:

- клієнт;
- робітник компанії з продажу автомобілів.

Клієнт, котрий має можливість заповнити побажання по майбутньому автомобілю та персональні дані.

Робітник компанії з продажу автомобілів має можливість власноруч заповнити дані клієнта при зверненні клієнта до компанії за телефоном, переглядати список автомобілів, вести список автомобілів, переглядати список клієнтів, знаходити рекомендації для клієнтів, редагувати дані клієнтів,

управляти процесом роботи з клієнтом. Структурну схему варіантів використання наведено в графічному матеріалі.

1.1.3. Структурна схема діяльності з підбору рекомендацій

Структурна схема діяльності наведена в графічному матеріалі.

Процес починається з заповнення клієнтом заявки для підбору автомобілів в веб-застосунку. Клієнт заповнює персональні дані, а також обирає бажані характеристики автомобіля. Після чого дані зберігаються в рекомендаційній системі. Робітник компанії з продажу автомобілів при перегляді списку клієнтів бачить, що з'явився новий клієнт і починає з ним роботу, змінює його статус. Після зміни статусу система знаходить рекомендації на основі побажань клієнта та даних по усім автомобілям в системі, що є доступними (мають статус Active). Після того, як система знайшла рекомендації, статус клієнта змінюється на відповідний. Робітник компанії з продажу автомобілів бачить, що статус клієнта змінився і автомобілі, що могли б підійти клієнту відображаються в таблиці на формі для цього клієнта. Робітник проводить комунікацію з клієнтом за телефоном чи email адресою, яку вказав клієнт при створенні заявки. Після того, як клієнт здійснює свій вибір, робітник компанії з продажу автомобілів заповнює поле обраного автомобіля, запис обраного автомобіля стає не активним для пошуку рекомендацій іншим клієнтам, система змінює статус запису клієнта на відповідний. Після цього йде завершення роботи з клієнтом. У разі, якщо клієнт придбав автомобіль, робітник компанії з продажу автомобілів змінює статус клієнта, автомобіль залишається недоступним для пошуку рекомендацій для інших клієнтів, створюється запис сутності контакт для збереження даних клієнта, що можуть бути використані для майбутніх маркетингових цілей. Якщо клієнт відмовляється від покупки автомобіля або робітник компанії з продажу автомобілів не може зв'язатись з клієнтом, то робітник змінює статус запису клієнта на відповідний, запис обраного автомобіля активується для того, щоб автомобіль використовувався для пошуку рекомендацій іншим клієнтам.

1.2. 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 завершення роботи зі зверненням клієнта, сутність Leads - інформацію про персональні дані клієнта під час звернення та переваги до вибору автомобіля, Leads_Cars – проміжна сутність для зв'язку сутностей Lead та Car, Cars- інформацію про автомобілі. Сутності Contacts та Leads мають зв'язок один або нуль до багатьох. Таблиця Lead та Cars мають зв'язок багато до багатьох та багато до одного або нуля.

Структурну схему бази даних наведено в графічному матеріалі. В таблиці 1.1. наведено опис сутностей та їх полів.

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	leadid	Унікальний ідентифікатор потенційного клієнта

	потенційних клієнтів	firstname	Ім'я клієнта
		Lastname	Прізвище клієнта
		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
		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
		ansbo_propulsiontype	Тип рушійної сили
		ansbo_doorscount	Кількість дверей
		ansbo_geartype	Тип приводу
		ansbo_price	Ціна
ansbo_consumption	Споживання палива		

1.3. Висновки до розділу 1

У розділі опису проектних рішень з розробки рекомендаційної системи підбору автомобілів для продажу клієнтам визначено мету дослідження, призначення системи, завдання, які необхідно виконати для досягнення мети. Також, описано бізнес-процес, розроблено структурну схему діяльності та IDEF0 діаграми для зображення структури бізнес-процесів, розроблено структуру рішення з інформаційного забезпечення у вигляді структурної схеми бази даних й описано сутності та поля у вигляді таблиці.

РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ ПОШУКУ РЕКОМЕНДАЦІЙ

2.1. Змістовна постановка задачі

Для звернення з метою вибору автомобіля клієнт заходить на сайт компанії, та заповнює характеристики автомобіля. Робітник компанії отримує заявку та власноруч проводить пошук автомобілів, які могли б підійти клієнту. Оскільки клієнт обирає характеристики на власний розсуд, то вони можуть суперечливими для конкретного автомобіля. Робітник компанії виконує пошук рекомендацій серед великої кількості автомобілів.

Необхідно знайти автомобілі з найбільшою кількістю характеристик, що відповідають побажанням клієнта серед наявних характеристик автомобілів, які пропонує компанія.

2.2. Математична постановка задачі

Дано:

1. Вектор характеристик, які обрав користувач $\bar{U} = \{u_0, u_1, \dots, u_n\}$, де u_j - певна характеристика, обрана користувачем, n - загальна кількість характеристик, $j = 0, 1, \dots, n$;

2. Вектор характеристик для певного автомобіля $\bar{I}_t = \{i_0, i_1, \dots, i_n\}$, 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 характеристик автомобілів, що мають найменший коефіцієнт подібності.

2.3. Огляд методів розв'язання

Всесвітня мережа інтернет розростається з кожним днем, збільшується кількість сайтів, сервісів, тощо. Окрім збільшення загальної кількості ресурсів

щодня та щохвилини відбувається збільшення обсягу даних на кожному сайті. Якщо розглядати такі сервіси, як інтернет магазини, то кількість товарів, що вони містять, постійно збільшується та залежить від розробки нових продуктів у всіх сферах життя. Користувач, що заходить в інтернет магазин і хоче обрати певний товар витрачає багато часу для досягнення своєї цілі. І обсяг часу, який потрібний для цього, щодня збільшується тому, що щодня асортимент магазинів поповнюється тисячами нових позицій. Аналогічна ситуація і з іншими веб ресурсами, що поповнюються терабайтами нової інформації. Рекомендаційні системи допомагають користувачам мінімізувати час на знаходження потрібної інформації, товарів, послуг.

Рекомендаційна система - система, що знаходить список предметів за певними критеріями, які б могли підійти користувачу. Для знаходження рекомендацій можуть використовуватися дані, які власноруч заповнює користувач, дані активностей користувача, дані профіля користувача та інша інформація про користувача.

Наразі у сфері ІТ використання рекомендаційних систем все більш стає популярним. Рекомендаційні системи використовують у сфері електронної торгівлі для знаходження товарів, що могли б зацікавити користувачів, в стримінгових системах для рекомендації цікавих ефірів, відео, каналів, в маркетингу для знаходження цільових груп користувачів. Також, браузерні використовують вбудовані в пошукові системи рекомендаційні системи для знаходження більш точних результатів по запиті користувача, фільтрації забороненого та спам контенту. Клієнти електронної пошти також використовують рекомендаційні системи для розбиття вхідних повідомлень на групи.

Рекомендації поділяють на два види:

- персоналізовані рекомендації;
- неперсоналізовані рекомендації.

Існують такі підходи знаходження персоналізованих рекомендацій, як фільтрація вмісту даних, колаборативна фільтрація, гібридна фільтрація, яка

суміщає два попередніх підходи.

Основною ідеєю підходів, що базуються на контенті, є знаходження схожих характеристик предметів, які обирає певний користувач для того, щоб рекомендувати користувачу предмети, які схожі на ті, яким користувач раніше ставив гарні оцінки. До недоліків підходів, що базуються на контенті можна віднести те, що вони погано справляються з знаходженням рекомендацій у випадку, яку рекомендаційна система, що використовує даний підхід має обмежений обсяг даних та невелику кількість характеристик для користувачів та предметів [5].

На відміну від підходів, що базуються на контенті, підходи колаборативної фільтрації знаходять рекомендації на основі оцінок про предмети від інших наявних в системі користувачів. Підхід базується на припущенні про те, що у двох користувачів мають бути схожі оцінки на певний предмет, якщо обидва користувача поставили високу оцінку одному і тому самому предмету. Також, підхід колаборативної фільтрації дозволяє урізноманітнити рекомендації за рахунок того, що кожний користувач має власні вподобання і у схожих користувачів може бути якась частина схожих рекомендацій, а якась зовсім різна. Але за рахунок таких вподобань користувач може отримати рекомендацію на предмет, яким він ще взагалі не цікавився, але може зацікавитись[5].

Методи колаборативної фільтрації поділяють на два типи:

- методи, які базуються на знаходженні сусідів;
- методи, які базуються на навчанні моделі.

Методи, які базуються на знаходженні сусідів використовують наявні оцінки в системі для користувачів та товарів. Дані методи поділяють на два типи: орієнтовані на користувача та орієнтовані на предмет. При використанні методів знаходження сусідів, що орієнтовані на користувача знаходяться товари, які могли б підійти певному користувачу на основі оцінок від інших користувачів для цих товарів. При використанні методів знаходження сусідів, що орієнтований на товар, предметом знаходження рекомендацій є сам користувач.

На основі характеристик певного товару знаходяться користувачі, що ставили гарні оцінки товарам, що є схожим на товар для якого відбувається пошук рекомендацій. Відмінністю цих методів є цільовий предмет знаходження рекомендацій, а спільною рисою є те, що вони знаходять рекомендації на основі наявних в системі оцінок користувачів. [5]

Методи, які базуються на навчанні моделі, використовують наявні дані в системі для навчання моделі. І потім на натренованій моделі знаходяться рекомендації. Але такий підхід потребує досить великий набір початкових даних для того, щоб процес знаходження рекомендацій був точним. Також, використовують практику перенавчання моделі з часом для збільшення точності знаходження рекомендацій. Цей підхід є можливим через те, що база даних системи весь час поповнюється новими оцінками від користувачів [5].

Рекомендаційні системи є дуже актуальними в наш час тому, що все більше сервісів і галузей починають їх використовувати, а тому існує багато наукових робіт за даною темою.

У статті [5] розглядається два методи створення рекомендаційних мереж: фільтрація вмісту і колаборативна фільтрація. Розглянуті методи базуються на аналізі профілів користувачів або поведінці користувачів. Більш детально розглядається колаборативний метод створення рекомендаційних мереж, основні кроки заснованого на сусідстві підходу та заснованого на моделях підходу колаборативної фільтрації. Також, було проведено порівняння розглянутих методів створення рекомендаційних мереж.

У статті [6] розглядаються різні підходи до побудови рекомендаційних мереж. Розглянуто підхід збору даних про користувача, фільтрацію вмісту та колаборативну фільтрацію. Також, у статті розглядаються та описуються основні проблеми рекомендаційних систем: проблема холодного старту, бульбашки фільтрів. При розгляді фільтрації вмісту розглядаються такі методи класифікації, як класифікатори на основі Баєсівських мереж, класифікатори на основі нейронних мереж, класифікатори на основі дерев рішень, класифікатори на основі алгоритмів кластеризації. Також, було розглянуто підходи до

колаборативної фільтрації. Під час розгляду колаборативної фільтрації було розглянуто методи обчислення коефіцієнтів подібності, як Евклідова відстань, Манхеттенська відстань та коефіцієнт кореляції Пірсона.

У статті [7] розглядаються рекомендаційні системи, що використовуються в соціальних мережах. Для знаходження рекомендацій використовуються моделі, що будуються у вигляді графу. Для знаходження рекомендованих публікацій для певних користувачів в рамках соціальної мережі використовується стохастичний граф, в якому вершинами є користувачі та публікації, а ребрами є дії користувачів відносно певних публікацій. Для знаходження рекомендацій відбувається аналіз побудованого графу.

У статті [8] розглядаються основні методи знаходження рекомендацій та більш детально описується гібридний метод знаходження рекомендацій. Він включає в себе колаборативну фільтрацію та фільтрацію на основі контенту. Також, в статті було розглянуто основні типи гібридних рекомендаційних систем:

- рекомендаційна система, що використовує метод колаборативної фільтрації з додатковими можливостями від метода фільтрації контенту;
- рекомендаційна система, що використовує метод фільтрації з додатковими можливостями від метода колаборативної фільтрації.

У статті [9] розглядається проблема відсутності можливості надавати рекомендації для користувачів, для яких немає історії користування системою (відсутні профілі). В даній статті було запропоновано використовувати гібридний підхід, що буде враховувати не лише вектори контентних характеристик предметів, а й вектори демографічних характеристик користувачів.

У статті [10] більш детально розглядається основна проблема рекомендаційних систем- проблема холодного старту. В статті зазначені основні причини виникнення даної проблеми, зазначено про можливу циклічність даної проблеми, а також наведено основні методи її вирішення:

- гібридизація рекомендаційної системи з використанням

колаборативної фільтрації та фільтрації на основі контенту;

➤ використання характеристик користувача під час знаходження рекомендацій (регіон, країна, дата та час, часова зона, тощо).

У статті [11] наведено класифікацію рекомендаційних систем, що відрізняється від класифікацій, які розглядалися в попередніх статтях. Автори наводять розділення рекомендаційних систем на дві категорії:

➤ рекомендаційні системи, що використовують алгоритми на основі пам'яті;

➤ рекомендаційні системи, що використовують алгоритми на основі моделі.

Алгоритми на основі пам'яті використовують всі існуючі дані оцінок та проводять пошук рекомендацій за допомогою зіставлення певного користувача з уже наявними даними інших користувачів.

Алгоритми на основі моделі використовують параметризовану модель. Для налаштування моделі використовують дані про оцінки інших користувачів. Процес знаходження рекомендацій відбувається на основі моделі.

До найбільш поширених алгоритмів на основі пам'яті відносять:

➤ алгоритми колаборативної фільтрації, що базуються на даних користувачів;

➤ алгоритми колаборативної фільтрації, що базуються на даних предметів.

Дані алгоритми знаходять рекомендації, використовуючи результати оцінок для найбільш схожих за параметрами користувачів або товарів. Алгоритми колаборативної фільтрації базуються на ствердженні, що якщо схожі користувачі за певним рядом параметрів дають схожу оцінку по одному предмету, то ці самі користувачі мають однаково оцінювати і інші предмети. Аналогічним чином дане ствердження працює і в іншому напрямку. Якщо два схожих предмета за певним рядом параметрів подобаються одному користувачу, то вони мають також подобатися і іншому користувачу.

Для алгоритмів на основі моделі основним принципом є початкова

побудова і тренування моделі. Після того, як модель натренована на тестових даних, вона може використовуватися для знаходження рекомендацій для інших користувачів. Тестовими даними є дані оцінок певних товарів від користувачів.

У статті [12] розглядається знаходження подібності між предметами за допомогою знаходження подібності між їх описами. Опис кожного предмета перетворюється на вектор слів та для кожного опису предмета розраховується вектор оберненої частоти (TF-IDF). Частотою є відносна частота певного терміну. Зворотна частота документа - відносна кількість векторів опису, елементом якого є це слово. Для побудови матриці використовується добуток відносної частоти певного слова до зворотної частоти документа. Побудова даної матриці потрібна для того, щоб слова, які найчастіше використовуються в описах предметів, мали менший вплив на знаходження рекомендацій ніж ті, які зустрічаються більш рідко, а отже допоможуть більш точно знайти найбільш схожі предмети. На основі отриманої матриці автори за допомогою косинусної подібності знаходять найбільш схожі предмети.

У статті [13] розглядається побудова ситуаційної моделі користувацького вибору. Рекомендації знаходяться на основі орієнтованого на схожості користувачів підходу. Модель являє собою дводольний багат шаровий граф. Кожний шар – переваги користувача в певний момент часу. Вершинами графу є користувачі та предмети. Зв'язок між шарами здійснюється між вершинами предметів. Кожний шар містить зв'язки між вершинами користувачів та вершинами предметів. Дуги, що сполучають вершини, залежать від явного або неявного зворотного зв'язку. У випадку, коли зв'язок задається на основі оцінки користувача, то він є явним. Якщо зв'язок задається на основі вибору користувача, то він є неявним. Важним аспектом запропонованого підходу є те, що багат шаровість графу дозволяє відслідковувати зміни виборів та оцінок клієнтів з зміною часу, та дозволяє аналізувати тенденції виборів, зміни вподобань для користувачів та надавати більш точні рекомендації.

У статті [14] розглянуто підходи до отримання початкових даних для знаходження рекомендацій. Основними підходами є:

- явний зворотній зв'язок;
- неявний зворотній зв'язок;
- гібридний зворотній зв'язок.

При використанні явного зворотного зв'язку система пропонує користувачеві через системний інтерфейс надати оцінки для предметів, щоб побудувати та вдосконалити свою модель. Точність рекомендацій залежить від кількості оцінок, наданих користувачем. Недоліком цього методу є те, що він вимагає зусиль від користувачів, а також користувачі не завжди готові надати достатньо інформації. Незважаючи на той факт, що явний зворотний зв'язок вимагає більше зусиль від користувача, він вважається як джерело отримання найбільш надійних даних, оскільки він не передбачає вилучення переваг, а також забезпечує прозорість процесу рекомендацій, що призводить до дещо вищого сприйняття рекомендацій, та збільшує рівень довіри до рекомендацій.

При використанні явного зворотного зв'язку система автоматично визначає уподобання користувача. Уподобання визначаються за допомогою відслідковування дій користувача: відслідковування історії покупок, історія перегляду товарів, аналіз переходів за посиланнями, аналіз пошукових запитів, аналіз вмісту електронної пошти, аналіз натиснення кнопок, тощо. Неявний зворотний зв'язок не потребує ручної оцінки товарів, а проводить аналіз переваг на основі історії дій в системі. Метод є менш точним у порівнянні з явним зворотнім зв'язком, оскільки може враховувати дії, які користувач виконав випадково, але в цей самий час такі уподобання є більш об'єктивними, оскільки переваги формуються автоматично на основі дій, і користувач не має змоги упереджено зменшити або збільшити рейтинг певного товару через якісь власні переконання.

Гібридний зворотний зв'язок містить явний та неявний зворотній зв'язок. Даний метод є найбільш гарним варіантом тому, що ручні оцінки товарів комбінуються з формуванням уподобань на основі історії дій. Даний тип зворотного зв'язку може бути налаштований відповідно до потреб систем. Наприклад, ручне проставлення оцінок певним товаром може бути не

обов'язкове, а використовуватись лише тоді, коли користувач вирішить висловити свою оцінку.

Також, у статті було запропоновано періодично проводити навчання системи на основі отриманих даних зворотного зв'язку, що покращить рівень знаходження рекомендацій.

У статті [15] розглядаються рекомендаційні системи з персональним ціноутворенням. Даний тип рекомендаційних систем передбачає відображення знижок на товари лише певним групам клієнтів. Клієнтів розділяють на такі групи:

- байдужі;
- чутливі до знижок;
- стандартні.

Клієнти, що відносяться до групи байдужих не купують товар зі знижкою або без знижки.

Клієнти, які входять в групу чутливих до знижок зазвичай купують товари лише зі знижками.

Клієнти, яких відносять до стандартної групи готові купувати товари за стандартною ціною без знижок.

Відповідно ціллю рекомендаційної системи є знаходження рекомендованої групи користувачів для клієнта на основі його характеристик. За допомогою такого розподілу досягається максимальний прибуток для власників рекомендаційної системи тому, що скидки отримують лише чутливі до знижок клієнти, а всі інші клієнти купляють товари за стандартною ціною.

У статті [16] розглядаються показники надійності рекомендаційних систем та методи ідентифікації атак на рекомендаційну систему.

Основним показником, що використовується при оцінці надійності рекомендаційної системи є зсув прогнозування. Зсув прогнозування - це показник, який показує на скільки змінилися рекомендації до і після атаки. Однак можуть бути випадки, коли оцінка для певних товарів є дуже низькою, і тому певний товар зі зміною оцінки під час атаки все одно не потрапляє в

рекомендації. Для збільшення точності оцінки надійності додатково використовують такий показник, як коефіцієнт звернень. Для оцінки надійності системи створюють дві вибірки: вибірку з реальними користувачами, вибірку з користувачами, що моделюють атаку. Розраховуються наведені показники, після чого проводиться аналіз і порівняння отриманих результатів.

Для виявлення атаки на рекомендаційну систему здійснюють пошук профілів ботів, потім їх видаляють з системи, а також всі дані, що вони залишили в системі: рекомендації, коментарі, оцінки. Для знаходження профілів ботів використовують класифікатор, що будується на основі алгоритмів машинного навчання.

У статті [17] розглядається побудова асоціативних правил для знаходження рекомендацій. Типову матрицю з користувачами та предметами, які вони оцінили, перетворюють на матрицю користувачів та їх інтересів. Для користувачів, які вже мають історію оцінок за асоціативними правилами, знаходять подальші рекомендації. Для нового користувача надають список з, наприклад, десяти предметів з найбільшим рейтингом оцінок від інших користувачів. На основі першого вибору користувача будуються асоціативні правила для подальших знаходжень рекомендацій.

У статті [18] розглядається знаходження рекомендацій для спільнот користувачів. Для знаходження рекомендацій для групи користувачів треба розділити всю множину користувачів, що містяться в певній системі, на групи схожих користувачів. Мірою схожості між користувачами є Евклідова відстань між векторами характеристик користувачів. Автори розглядають використання метода k -середніх для кластеризації множини користувачів. Недоліками методу k -середніх є те, що він локально збіжний, потребує задання початкової кількості кластерів та центрів кластерів. Після розбиття множини на кластери розраховується сумарна рейтингова оцінка для предметів від цієї групи користувачів. Після потрапляння нового клієнта знаходять групу користувачів на яких він максимально схожий і надають рекомендації на основі рейтингу сумарних оцінок.

У статті [19] розглядається модифікація алгоритму колаборативної фільтрації з використанням релевантностей. Було порівняно класичний алгоритм оцінки релевантності товарів з модифікованим. При класичному алгоритмі враховується показник купив або не купив товар, відвідав або не відвідав сторінку з товаром, тощо. Автори пропонують замість бінарного показника використовувати час, який клієнт провів на сторінці товару, як міру зацікавленості користувача певним товаром. Якщо час більше якогось порогового значення, то товар є релевантним, інакше нерелевантним. Використання даного показника забезпечує більш точне знаходження рекомендацій, оскільки замість лише куплених товарів система буде пропонувати схожі товари до тих, які користувач переглядав довше всього.

У статті [20] розглядається рекомендаційна система, що враховує емоційне забарвлення коментарів користувачів. При побудові рекомендаційних систем часто використовують лише позитивні відгуки. Користувачі ставлять мітку сподобалось або нічого не ставлять. Даний підхід втрачає відгуки від багатьох користувачів тому, що користувач не хоче ставити мітку сподобалось і не ставить нічого взагалі, але він може написати коментар, тому аналіз емоційного забарвлення коментарів є досить непоганим показником у сполученні з оцінками або у разі відсутності оцінок взагалі. Для того, щоб заповнити пусті оцінки для певних предметів, емоційне забарвлення відгука чи коментаря буде перетворюватися в оцінку за допомогою факторизаційної машини, а контекстну інформацію відгука використовувати, як додатковий параметр для факторизаційної машини.

У статті [21] розглядається рекомендаційні системи, що будуються на основі онтологій. Онтології - це спільні концепції предметних галузей. За допомогою онтологій формалізують певну предметну галузь, розбиваючи її на певний список об'єктів та зв'язків між ними. Для знаходження рекомендацій рекомендаційна система робить кон'юнктивний запит до наявних онтологій та намагається знайти предмети, які задовольняють всім обмеженням, які обрав користувач. У випадку, коли не знайдено жодного предмета, система здійснює

запит до списку онтологій зі зменшенням кількості обмежень, які беруться зі списку обмежень наданих клієнтом. Система поступово зменшує кількість обмежень до того моменту поки не знайде принаймні один предмет, що задовольнить обмеженням. Після знаходження результату у разі, якщо рекомендації було знайдено не по всім обмеженням, система виводить повідомлення користувача, що деякі обмеження не враховувались в пошуку та запитує чи відобразити результат. У разі згоди клієнта список рекомендацій відображається, а також наводиться список обмежень, які були виключені для знаходження рекомендацій. Користувач може змінити дані обмеження і повторити пошук.

У статті [22] розглядається рекомендаційна система, що має містити два рівні знаходження рекомендацій. Рекомендаційна система має знаходити предмети на першому рівні, а потім на основі знайдених рекомендацій, рекомендувати розподіл користувачів в групи, що могли б бути зацікавлені в предметах, що знайдені на першому рівні. Для вирішення задачі знаходження обмежень для розподілу користувачів на групи пропонується підхід, в якому обмеження по розподілу користувачів задаються як нечіткі множини, після чого використовують фаззифікацію для перетворення нечітких множин обмежень в чіткі параметри для розподілу користувачів на групи.

У статті [23] розглядається спосіб зберігання великої кількості даних в рекомендаційних системах. Оскільки алгоритми знаходження рекомендацій зазвичай використовують та обробляють великі обсяги даних, то необхідно мінімізувати час виконання запитів для сховища даних тому, що час роботи алгоритмів напряду залежить від часу обробки інформації. Автори пропонують використовувати NoSQL бази даних. Серед переваг NoSQL баз даних є те, що вони мають набагато більшу продуктивність, близьку до лінійної масштабованості, відсутність чіткої схеми даних. Також, до переваг таких баз даних слід віднести вбудовану паралелізацію запитів.

У статті [24] розглядається побудова рекомендаційної системи з даними користувачів, що не є частиною бази даних рекомендаційної системи. Автори

пропонують використовувати профілі соціальних мереж користувачів для аналізу вподобань. Даний підхід забезпечить менший обсяг бази даних системи, а також вирішує проблему наповнення даними профілю користувача. Але такий підхід потребує постійної доступності серверів соціальних мереж, а також попереднього збору даних по користувачам перед пошуком рекомендацій.

В результаті аналізу публікацій виявлено, що у розглянутих публікаціях не було розглянуто ситуацію, коли для пошуку рекомендацій потрібно на основі побажань користувача надати рекомендацію серед існуючих об'єктів. Дана ситуація є актуальною, оскільки існує багато випадків, коли користувач не має профіля на деякому ресурсі, приходять вперше до місця надання певних послуг, але хоче отримати рекомендацію для, наприклад, придбання деякого товару, вибору туристичної путівки, вибору фільму в кінотеатрі, отримання деякої послуги, тощо. Тому обраний напрям дослідження і створення рекомендаційної системи є актуальним.

2.4. Розробка алгоритму пошуку рекомендацій

Для побудови рекомендаційної системи використано підхід фільтрації вмісту даних [5] відповідно до особистих запитів клієнта. Елементами в алгоритмі пошуку рекомендацій вважаються автомобілі, наявні в системі, користувачами - запити клієнтів, що містять бажані характеристики автомобілів.

Першим етапом роботи алгоритму є перетворення текстових характеристик бажаних параметрів для автомобілів, отриманих з заявки клієнта, та характеристик автомобілів на вектори слів, індексація та нормалізація отриманих векторів. Перший етап роботи алгоритму наведено кроками 1-5 на блок-схемі роботи алгоритму, яка наведена в графічному матеріалі [4].

Другим етапом роботи алгоритму є саме пошук рекомендацій. Процес пошуку рекомендацій відбувається до тих пір поки не знайдена кількість автомобілів, які відповідають запиту. Другий етап роботи алгоритму наведено кроками 6-12 на блок-схемі роботи алгоритму, яка наведена в графічному матеріалі [4].

Для знаходження списку елементів (автомобілів), що підходять користувачеві, фільтруємо елементи, що описані векторами атрибутів таким чином, щоб вони задовольняли перевагам конкретного користувача. У цьому випадку основна ідея алгоритму полягає в тому, щоб знайти схожість між об'єктом - користувачем та елементами, що мають різну семантику, але перебувають у тому самому Евклідовому n -вимірному просторі. Кореляція між об'єктом - користувачем та векторами атрибутів елементів представлена як відстань. Показник подібності, представлений Евклідовою відстанню між атрибутами запитів користувачів та атрибутами елементів, обчислюється на основі найбільш подібних атрибутів (наприклад, що мають найближчі відстані між собою) [6]. Тобто, отримуємо схожість між векторами побажань користувачів та векторами характеристик автомобілів, обчислюючи суму квадратів часткових відстаней між окремими атрибутами.

Алгоритм знаходить елементи, що найбільше підходять конкретному елементу- користувачеві, за вектором атрибутів \bar{U} (вектором переваг користувача), та формує центр кластеру, в якому конкретний користувач, пов'язаний з кількістю елементів. Подібність користувачів та елементів представлена Евклідовою відстанню між векторами атрибутів користувача \bar{U} та атрибутів елемента \bar{I} .

Кількість атрибутів вектору користувача та векторів автомобілів є однаковою. Подібність знаходимо за формулою:

$$S_w^2(\bar{U}, \bar{I}_t) = \sum_{n=1}^{N_p} \left(\frac{1}{r_n} \cdot [i_n^{(t)} - U_n]^2 \right), \quad (2.1.)$$

де $S_w^2(\bar{U}, \bar{I}_t)$ - 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 -им атрибутом з вектору атрибутів користувача та вектором елементів,

N_p - загальна кількість атрибутів користувачів та елементів.

На відміну від звичайної Евклідової формули відстані, яка дозволяє обчислити фактичну відстань між двома векторами атрибутів, коефіцієнт подібності отримано у вигляді зваженого значення відстані, що розраховується, як сума квадратів часткових відстаней між конкретними атрибутами користувача або елемента, помноженого на певний ваговий коефіцієнт r_n . Тобто знаходиться подібність, що базується на відстані та значення якої залежить від більш ніж одного параметра.

У цьому випадку, крім часткової відстані між атрибутами користувача та елемента, використано параметр - лексикографічна подібність 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 (2.1.)

$$r_n = \sum_j \left(\frac{1}{\min\{L\}} \left| i_j^{(n)} - u_j^{(n)} \right| \right), \quad (2.2.)$$

де r_n - коефіцієнт лексикографічної подібності між n -им 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 -ий атрибут користувача або елемента.

Для обчислення подібності атрибутів користувача або елемента будемо використовувати алгоритм лінійного пошуку шляхом ітерації над двома рядками, що представляють n -ий атрибут, і для кожної позиції виконуємо порівняння символів, розташованих на одній j -ій позиції. Якщо ці символи $i_j^{(n)}$ і $u_j^{(n)}$ збігаються, то ми додаємо значення $\frac{1}{L}$ до значення подібності r_n . Параметр L - це фактична довжина найменшого рядка, яка представляє n -ий атрибут користувача або елемента.

Крім того, для забезпечення більш гнучких обчислень подібності на відстані за допомогою формул (2.1., 2.2.) модифіковано формулу (2.1.), використовуючи інший додатковий параметр m , який представляє фактичну кількість атрибутів користувача або елемента, які 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 (2.1) матиме вигляд:

$$S_w^2(\bar{U}, \bar{I}_t) = \frac{1}{m} \cdot \sqrt{\sum_{n=1}^{N_p} \left(\frac{1}{r_n} \cdot [i_n^{(t)} - U_n]^2 \right)}, \quad (2.3.)$$

Велика кількість атрибутів з лексикографічною подібністю гарантує, що Евклідова відстань між двома векторами атрибутів користувача або елемента є невеликою, що забезпечує збільшення подібності між конкретним користувачем і елементом. Однак існують випадки, коли конкретний користувач і деякий елемент не мають атрибутів, які є подібними та значення додаткового параметра $m = 0$. У цьому випадку будемо приймати значення параметра $m = 0,01$ для забезпечення великої відстані між кардинально різними користувачем та елементом, щоб уникнути випадків, коли елементи з атрибутами, що зовсім не відповідають атрибутам користувача, мають ближчу відстань до цього користувача [3]. Блок-схему роботи алгоритму наведено в графічному матеріалі.

2.5. Приклад роботи алгоритму

Алгоритм на вхід приймає текстовий опис побажань користувача та текстові описи характеристик автомобілів. Приклад тестового опису побажань користувача, отриманого з системи, наведено у таблиці 2.1, а приклад тестового опису характеристик автомобілів, отриманого з системи, наведено у таблиці 2.2.

Таблиця 2.1.

Текстовий опис побажань користувача, отриманих з системи

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

Таблиця 2.2.

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 проводиться розбиття текстових описів характеристик на вектори характеристик, проводиться індексація слів та нормалізація отриманих індексів. У таблиці 2.3. наведено знайдені значення атрибутів вектору характеристик для автомобіля Audi A8.

Таблиця 2.3.

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 таблиці 2.4 наведено індекси атрибутів вектору характеристик для автомобіля Audi A8 після нормалізації

Таблиця 2.4.

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 таблиці 2.5 наведено знайдені індекси атрибутів вектору характеристик для автомобіля BMW I7.

Таблиця 2.5.

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 таблиці 2.6 наведено індекси атрибутів вектору характеристик для автомобіля Audi A8 після нормалізації

Таблиця 2.6.

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 вже існує такий самий атрибутів у векторі автомобілів. Якщо такий самий атрибут знайдено, йому присвоюється значення індексу вже існуючого елемента. У випадку, якщо такого самого значення немає, в векторах характеристик автомобіля проставляється індекс атрибута в векторі характеристик, після чого проводиться нормалізація індексів. У таблиці 2.7. наведено знайдені індекси атрибутів вектору побажань користувача.

Таблиця 2.7.

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 таблиці 2.8 наведено знайдені індекси атрибутів вектору побажань користувача після процесу нормалізації.

Таблиця 2.8.

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 завершення підготовки даних для вектору атрибутів побажань користувача знаходиться зважений коефіцієнт подібності з кожним вектором атрибутів характеристик автомобілів. Знаходиться автомобіль з найменшим значенням зваженого коефіцієнта подібності. Даний процес повторюється допоки не буде знайдено число автомобілів, що відповідає необхідній кількості рекомендацій. В нашому випадку необхідно знайти одну рекомендацію, тому автомобіль, що має менший зважений коефіцієнт подібності, є рекомендованим автомобілем. В таблиці 2.9. наведено розрахунок коефіцієнта лексикографічної подібності, добуток коефіцієнта лексикографічної подібності до квадрату різниці значень відповідних значень атрибута побажань користувача та вектору характеристик автомобіля Audi A8, сумарне накопичувальне значення для всього вектору автомобіля.

Таблиця 2.9.

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

Атрибут побажань користувача	Атрибут автомобіля	$\frac{1}{r_n}$	$\frac{1}{r_n} \cdot [i_n^{(t)} - U_n]^2$	$\sum_{n=1}^{N_p} (\frac{1}{r_n} \cdot [i_n^{(t)} - U_n]^2)$
------------------------------------	-----------------------	-----------------	---	--

1	2	3	4	5
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:

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

В таблиці 2.10. наведено розрахунок коефіцієнта лексикографічної подібності, добутку коефіцієнта лексикографічної подібності до квадрату різниці значень відповідних значень атрибута побажань користувача та вектору характеристик автомобіля BMW I7, сумарне накопичувальне значення для всього вектору автомобіля.

Таблиця 2.10.

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

Атрибут побажань користувача	Атрибут автомобіля	$\frac{1}{r_n}$	$\frac{1}{r_n} \cdot [i_n^{(t)} - U_n]^2$	$\sum_{n=1}^{N_p} \left(\frac{1}{r_n} \cdot [i_n^{(t)} - U_n]^2 \right)$
1	2	3	4	5
bmw	bmw	1	0	0
V8	V8	1	0	0
less_than_thirty	less_than_forty	0.6666666666666666	0.168265378539868	0.1682653785398682

		663	62	
Hybrid	Hybrid	1	0	0.16826537853986862
two	two	1	0	0.16826537853986862
full	back	0	0.112176919026579	0.28044229756644762
less_than_fifteen	less_than_fifteen	1	0	0.28044229756644762

Знаходимо коефіцієнт подібності для вектору характеристик автомобіля BMW I7:

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

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

2.6. Результати експериментальних досліджень

Було досліджено час роботи алгоритму при збільшенні кількості автомобілів, серед яких необхідно знайти рекомендації. На рисунку 2.1. зображено графік залежності часу роботи алгоритму від кількості автомобілів серед яких знаходяться рекомендації.

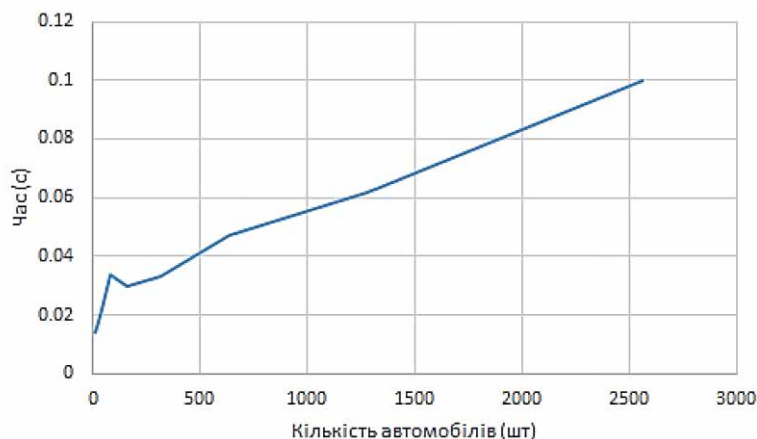


Рис. 2.1. Графік залежності часу роботи алгоритму від кількості автомобілів

З графіка видно, що час роботи алгоритму поступово збільшується. Однак

при збільшенні кількості наявних автомобілів в 25 разів час роботи час збільшився в 5 разів. Розглядаючи показник в 2500 автомобілів алгоритм працює 0,1 секунди, що є дуже гарним результатом. Дані результати свідчать про те, що алгоритм задовольняє показник масштабованості для рекомендаційних систем, а це свідчить про те, що з плином часу пошуку рекомендацій не буде займати багато часу.

Також, було досліджено залежність середнього значення міри подібності між вектором побажань користувача та векторами рекомендованих автомобілів від кількості автомобілів серед яких знаходяться рекомендації. На рисунку 2.2 зображено фрагмент графіку залежності середнього значення коефіцієнта подібності між вектором атрибутів користувача та векторами атрибутів рекомендованих елементів від загальної кількості векторів характеристик автомобілів марки BMW в системі.

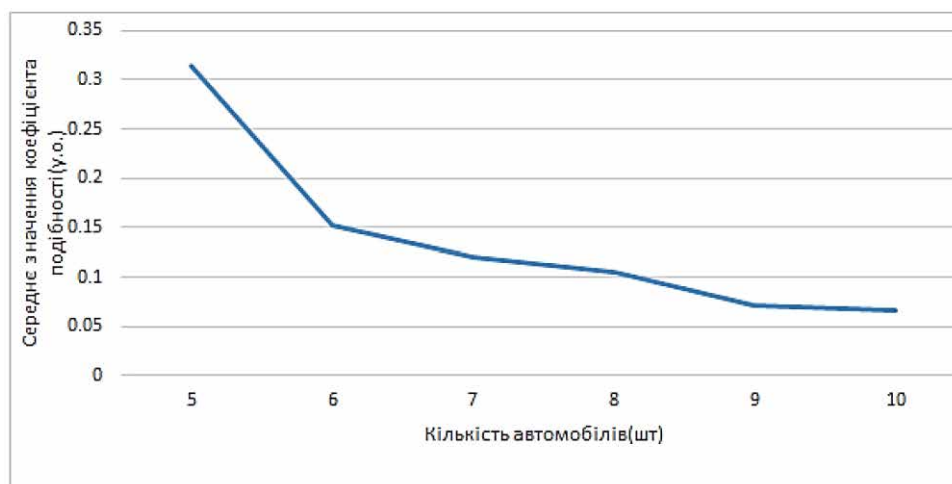


Рис. 2.2. Фрагмент графіку залежності середнього значення коефіцієнта подібності від загальної кількості автомобілів марки BMW в системі

Як видно з фрагменту графіку, зі збільшенням кількості наявних автомобілів середня відстань від вектору побажань клієнта до векторів характеристик автомобілів зменшується, а це свідчить про те, що зі збільшенням кількості автомобілів в системі, користувач буде отримувати рекомендаційний список, який буде складатися з найбільш підходящих автомобілів.

Також було досліджено залежність середнього значення коефіцієнта подібності між вектором атрибутів користувача та векторами атрибутів

рекомендованих елементів від кількості атрибутів в векторах характеристик автомобілів. Графік даної залежності зображено на рисунку 2.3.

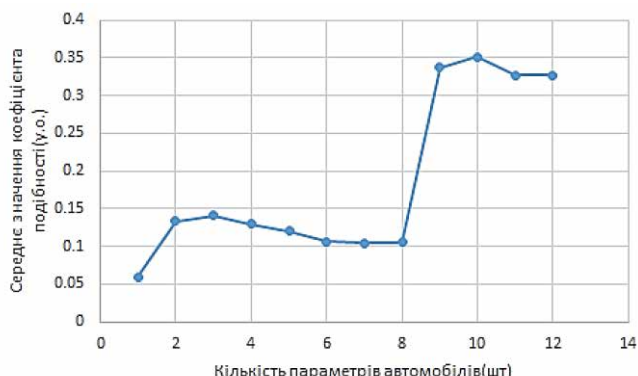


Рис. 2.3. Графік залежності середньої середнього значення коефіцієнта подібності від кількості атрибутів в векторах характеристик автомобілів

З графіку видно, що після восьми параметрів відбувається значний стрибок в середній відстані це означає, що при дев'яти атрибутах в векторах автомобілів знайдені автомобілі, які будуть рекомендуватися клієнту будуть більш різні і менше задовольняти потребам клієнта. Тому використовуємо вісім параметрів в векторах атрибутів характеристик автомобілів при пошуку рекомендацій.

Розроблений алгоритм можна використовувати при знаходженні рекомендацій тому, що він є стійким до збільшення даних, а тому задовольняє показнику масштабованості для рекомендаційних систем, також при збільшенні кількості автомобілів в системі користувач буде отримувати рекомендаційний список, який буде складатися з найбільш підходящих автомобілів.

2.7. Висновки до розділу 2

В розділі з моделей та методів пошуку рекомендацій була сформульована змістовна та математична постановки задачі пошуку рекомендацій, проведено огляд існуючих методів пошуку рекомендацій, розроблено та описано алгоритм, розроблено блок-схему алгоритму, наведено приклад роботи алгоритму та результати експериментальних досліджень. В ході огляду існуючих методів було виявлено, що в жодній публікації не було розглянуто ситуацію, коли для пошуку рекомендацій потрібно, на основі побажань користувача, надати рекомендацію

серед існуючих об'єктів, тому обраний напрям дослідження і пошуку рекомендаційної системи є актуальним.

РОЗДІЛ 3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Засоби розробки

Для створення фронт частини веб-застосунку було використано фреймворк Bootstrap, бібліотеку JQuery, яка побудована на мові програмування JavaScript. Для розробки серверної частини веб-застосунку використовувався ASP.Net Core, який використовує мову C#. Оскільки серверна частина веб-застосунку створює записи в Dynamics CRM, то для комунікації з CRM використовувалось API. Для створення HTTP-клієнта використовувався Nuget [26] пакет Duxix [29]. Системою для роботи з даними клієнтів, управління даними, опрацюванням запитів було обрано Dynamics 365 ForSales. Для створення обробника подій використовувався Net 4.6.2. з використанням інтерфейсу IOrganisationService [28] для виконання основних дій з сутностями. Для збереження даних було обрано Common Data Service(CDS).

C# - мова програмування, що використовує компілятор - для перетворення коду в IL код, після чого цей код перетворюється в двійковий. Даний підхід дозволяє знаходити всі синтаксичні помилки на етапі побудови та компіляції проекту, що є перевагою в порівнянні до тих мов, що використовують інтерпретатор. C# є однією з мов, які входять в платформу Net. Основним інструментом, який використовують Net розробники, є Microsoft Visual Studio. Дана IDE містить дуже великий обсяг різного функціоналу: відладка коду, трасування коду, відслідковування продуктивності та затрати ресурсів, публікація застосунків в Azure, доповнення функціоналу за власними потребами через розробку розширень, тощо. C# часто отримує оновлення від компанії Microsoft, а також розробники часто публікують власні бібліотеки, як nuget пакети і ці бібліотеки можна використовувати безкоштовно навіть в комерційних цілях. Також C# використовують в різних сферах розробки: Web, мобільна розробка, розробка десктопних застосунків, розробка хмарних

застосунків, розробка ігор.

JavaScript - це динамічна мова програмування, що не використовує строгу типізацію, тобто об'єкт може мати один тип, а після операції присвоєння може стати зовсім іншим. Також, JavaScript не може працювати з пам'яттю, але основним призначенням даної мови є робота з веб сторінками та браузером, тому робота з пам'яттю не є необхідною. За допомогою JavaScript в основному розробляють скрипти для динамічної зміни HTML розмітки, розробка анімацій для елементів сторінки, зміна обов'язковості полів, розрахунків певних значень на основі даних сторінок, виклику методів API. JavaScript використовує інтерпретатор, тому при розробці скриптів слід уважно перевіряти розробки та обов'язково проводити гарно тестування розробок тому, що помилки будуть виникати лише в момент виконання певного рядочку коду. Виклик методів API можна реалізовувати через Ajax, який забезпечує оновлення даних без перезавантаження сторінки.

jQuery - бібліотека побудована на мові JavaScript. Дана бібліотека полегшує та пришвидшує написання скриптів, через велику кількість функціоналу, що має базові реалізації. Також, бібліотека значно полегшує розробку анімацій через те, що вбудований функціонал для цього підтримується більшістю браузерів, що нівелює необхідність модифікації скрипта для роботи в кожному окремому браузері. Також, jQuery полегшує написання обробників подій на сторінках, динамічну зміну HTML розмітки, написання Ajax запитів.

Bootstrap - фреймворк, що використовує HTML, CSS, Javascript. Даний фреймворк містить велику кількість вбудованих CSS класів, javascript функцій, та HTML елементів, що в сукупності дозволяє пришвидшити процес розробки сторінок використовуючи вже готові елементи. Великою перевагою використання даного фреймворка є те, що всі елементи є адаптивними та працюють в більшості браузерів. Також, найчастіше фреймворк окрім кастомізації елементів використовують для побудови сіток. Даний фреймворк дає дуже гнучкі інструменти для їх побудови, тому він може задовольнити потреби будь-якого розробника. Для використання даного фреймворка

достатньо завантажити компоненти, розмістити в необхідну папку та підключити до сторінки або можна використовувати посилання на онлайн версії цих бібліотек, але в такому разі у випадках недоступності серверів, що містять дані бібліотеки, функціонал, що використовує Bootstrap не буде працювати.

Dyrix - NuGet пакет, що використовується для побудови HTTP клієнта для взаємодії з Dynamics CRM. Для підключення до CRM через API треба спочатку отримати токен, а потім використовуючи даний токен робити запити. Максимальний час дії токена - 1 година. Перевагою використання даного пакета є те, що він автоматично перевіряє валідність токена і у разі закінчення часу токена, пакет автоматично запитує новий і використовує його для наступних запитів.

NuGet - це менеджер пакетів для .NET. Вбудованими засобами клієнта NuGet можна публікувати розробки в галерею пакетів та завантажувати та використовувати розробки інших людей. Галерея NuGet - це загальне сховище, що містить всі пакети та використовується розробниками пакетів та людьми, що використовують дані пакети.

Dynamics 365 - сімейство продуктів, що включає в себе рішення ERP та CRM рішення. Під час розробки рекомендаційної системи було взято один із різновидів Dynamics CRM рішень, а саме Dynamics 365 for Sales, як систему для управління даними автомобілів, клієнтів, управління процесу по роботі з клієнтом, а також в Sales буде інтегруватись алгоритм для пошуку рекомендацій. На рисунку 3.1. зображено структуру сімейства продуктів Dynamics 365.

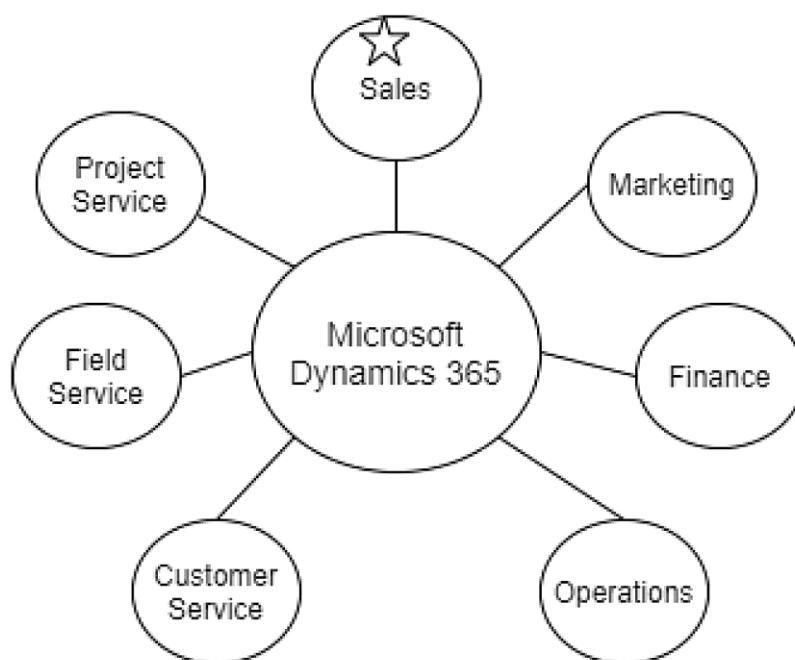


Рис. 3.1. Структура сімейства продуктів Dynamics 365.

Microsoft Dynamics 365 for Sales – модуль Dynamics 365, що відповідає за продажі та дозволяє збільшувати обсяги продажів за рахунок зберігання даних клієнтів та побудови індивідуального підходу до кожного клієнта на основі існуючих даних. Microsoft Dynamics 365 for Sales дозволяє здійснювати контроль за ходом роботи з клієнтами, отримувати статистику різного роду на основі даних, використовувати дані клієнтів у маркетингових цілях.

Інтерфейс `IOrganizationService` використовується CRM-розробниками, що працюють з Dynamics 365. Даний інтерфейс має набір методів для роботи з кастомними та системними сутностями. Інтерфейс дозволяє виконувати такі операції, як створення, оновлення, видалення, отримання одного запису, отримання багатьох записів, виконання системних запитів, виклики кастомних процесів, дозволяє отримувати метадані сутностей.

CDS (Common Data Service) - сервіс для роботи з даними. Використовуючи даний сервіс розробники працюють не з таблицями в базах даних, а з сутностями, що є об'єктами, легко налаштовуються через інтерфейс. Сутності в середині CDS перетворюються в записи таблиць БД. Перевагою CDS є те, що він може використовуватися в різних програмних застосунках компанії Microsoft.

3.2. Архітектура програмного забезпечення

3.2.1. Структурна схема класів

Розглянемо діаграму класів для плагіна, що спрацьовує на оновленні записів сутності клієнт (Lead). Діаграму класів для плагіна зображено на рисунку 3.2.

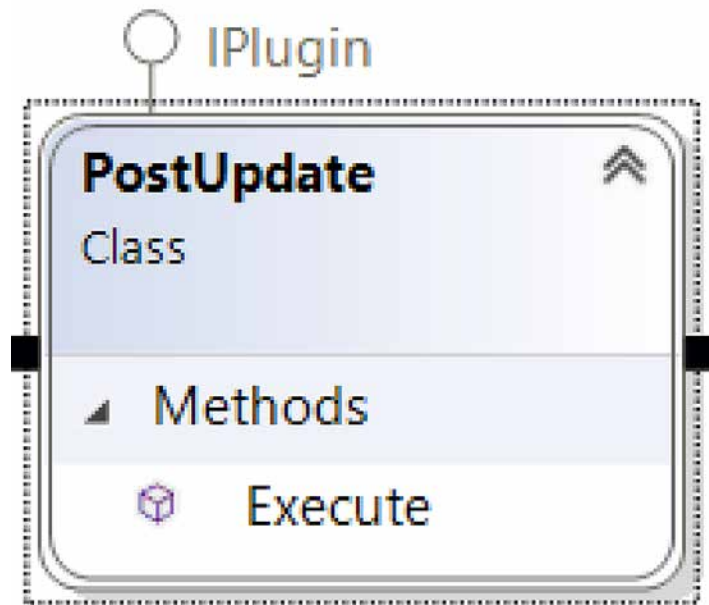


Рис. 3.2. Структурна схема класів плагіна

Плагін представлено класом **PostUpdate**. Даний клас реалізує інтерфейс **IPlugin** та містить метод **Execute**. Інтерфейс **IPlugin** - інтерфейс, що використовується усіма плагінами, що опубліковані чи будуть публікуватися в Dynamics CRM. Використання даного інтерфейсу забезпечує, що кожен плагін буде містити метод **Execute**. В методі **Execute** розміщується основна логіка плагіна, а також отримання контексту виклику, об'єкта, що містить всі зміни, сервісу трасування.

Для роботи з об'єктами було розроблено класи розширення. Класи розширення були розроблені для класу **Car**, **Entity**, інтерфейсу **IEnumerable**. Діаграму класів розширень наведено на рисунку 3.3.

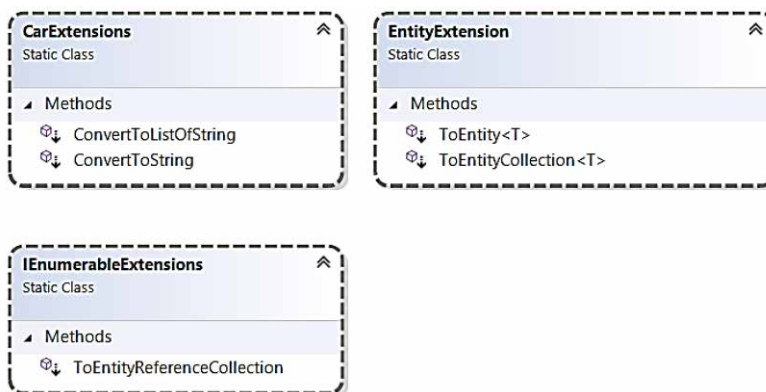


Рис. 3.3. Структурна схема класів розширень

Класи розширення приймають першим параметром екземпляр класу, для якого буде використовуватися, після чого можуть використовуватися додаткові параметри за потребою. 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` було обрано підхід `EarlyBound`. Тобто спочатку створюються моделі на основі відповідних полів в `Dynamics CRM`, а потім дані моделі використовуються для роботи з `CRM`. Діаграму класів для взаємодії з `CRM` зображено на рисунку 3.4.

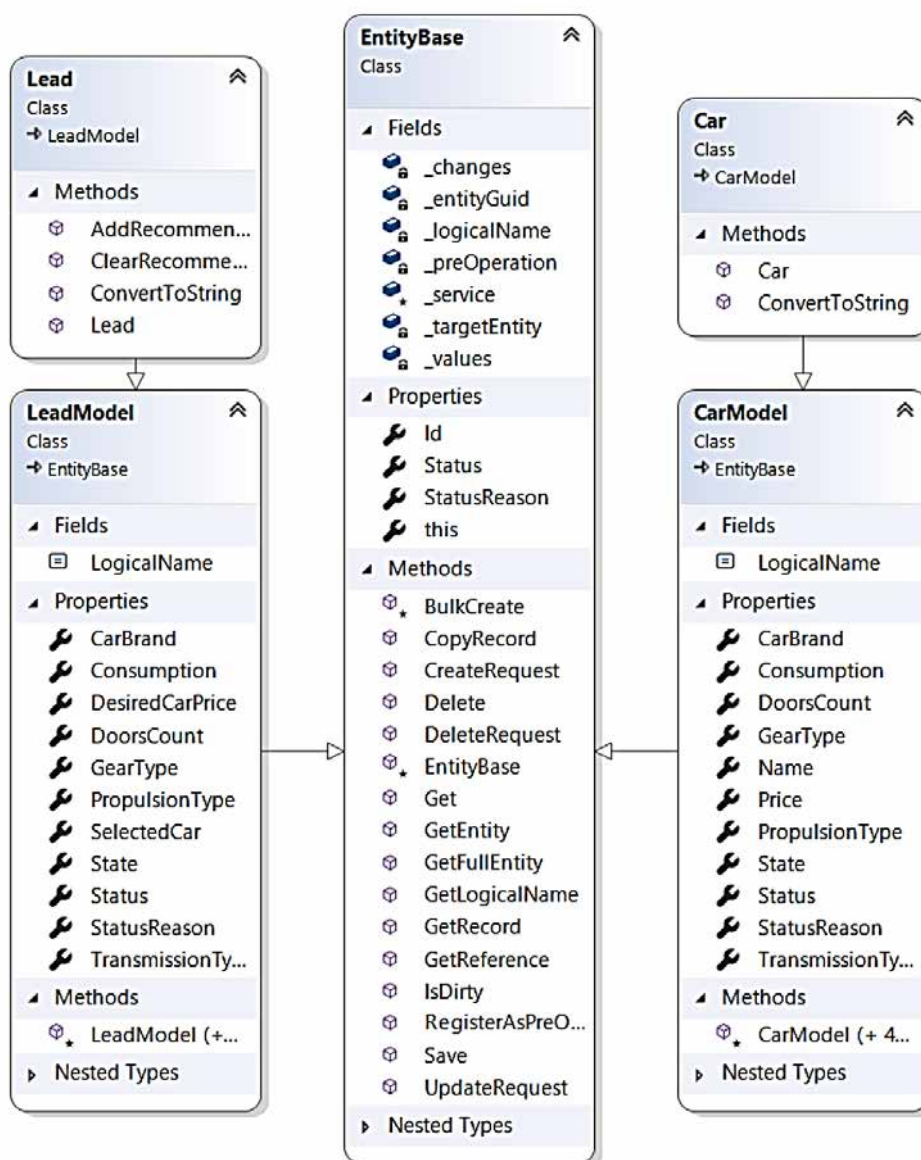


Рис. 3.4. Структурна схема класів для взаємодії з CRM

На діаграмі зображено 2 класи EntityProvider-а Lead та Car. Класи EntityProvider-и успадковуються від EarlyBound моделей LeadModel та 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 - EarlyBound модель та містить необхідні параметри з CRM у вигляді властивостей.

Клас CarModel - EarlyBound модель та містить необхідні параметри з CRM у вигляді властивостей.

Класи LeadModel та CarModel успадковуються від класу EntityBase.

EntityBase - базовий клас від якого успадковуються всі моделі. Він 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 знаходження рекомендацій в середині плагіну було розроблено клас RecommendationSearchService. Об'єкт даного класу створюється під час кожного виклику плагіна. Даний клас містить список автомобілів, що отримується з наявних записів в CRM, об'єкт користувача, мінімальне та максимальне значення для нормалізації. Також, для роботи пошуку рекомендацій було створено додаткові моделі. Структурна схема класів для знаходження рекомендацій зображено на рисунку 3.5.

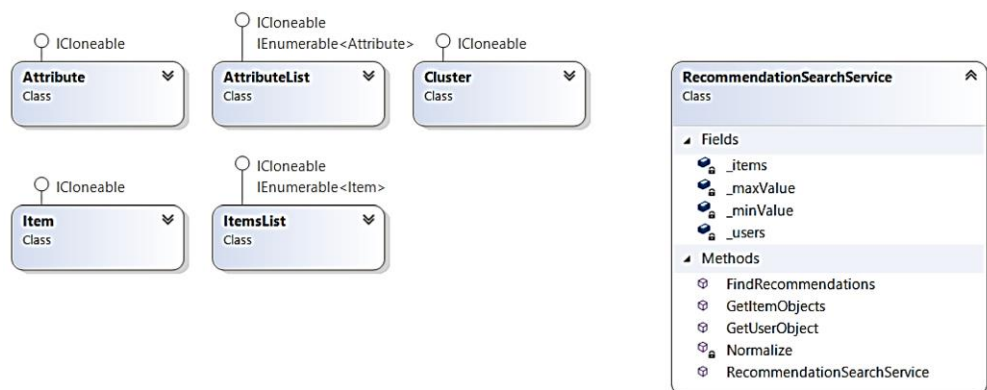


Рис. 3.5. Структурна схема класів для знаходження рекомендацій

Клас Item - клас елемента користувача або автомобіля, клас ItemList - список класів користувачів або автомобілів, Attribute - клас певної характеристики або побажання, AttributeList - список класів характеристик або

побажань, Cluster - клас групи елементів.

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

Для конвертації значенням, що отримуються із CRM системи, було розроблено інтерфейс IConverter, що забезпечить, що всі класи, які реалізують даний інтерфейс будуть мати метод Convert. Класи перетворювачі реалізують інтерфейс IConverter та мають єдиний метод Convert. Також, було створено клас ConversionService, який має метод Convert, що приймає параметром назву класу перетворювача ініціалізує інтерфейс IConverter та викликає метод Convert відповідного перетворювача. Діаграму класів для перетворення даних з CRM в текстові дані зображено на рисунку 3.6.

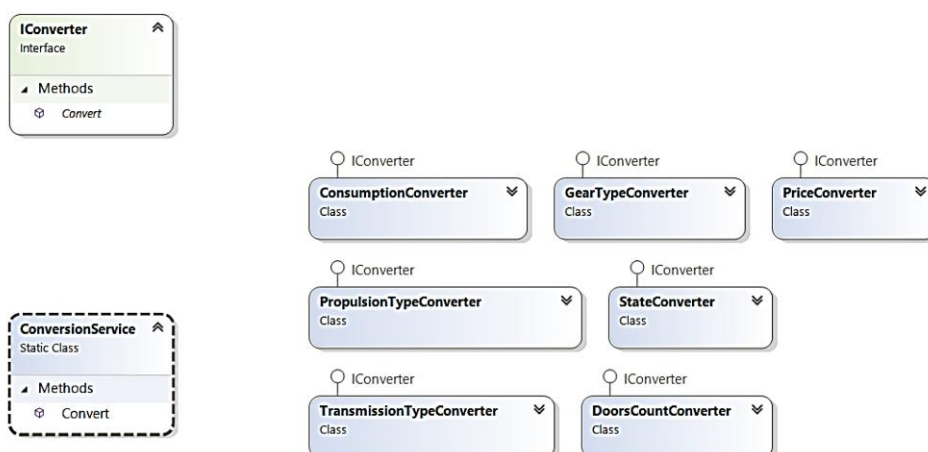


Рис. 3.6. Структурна схема класів для перетворення даних з CRM в текстові дані

3.2.2. Структурна схема послідовності

Структурну схему послідовності наведено в графічному матеріалі.

Клієнт працює лише з інтерфейсом веб застосунку. Інтерфейс передає дані до сервера, що в свою чергу передає дані до CRM системи, а та до CDS. У разі успішного збереження даних клієнт отримує повідомлення про те, що його запит було успішно прийнято.

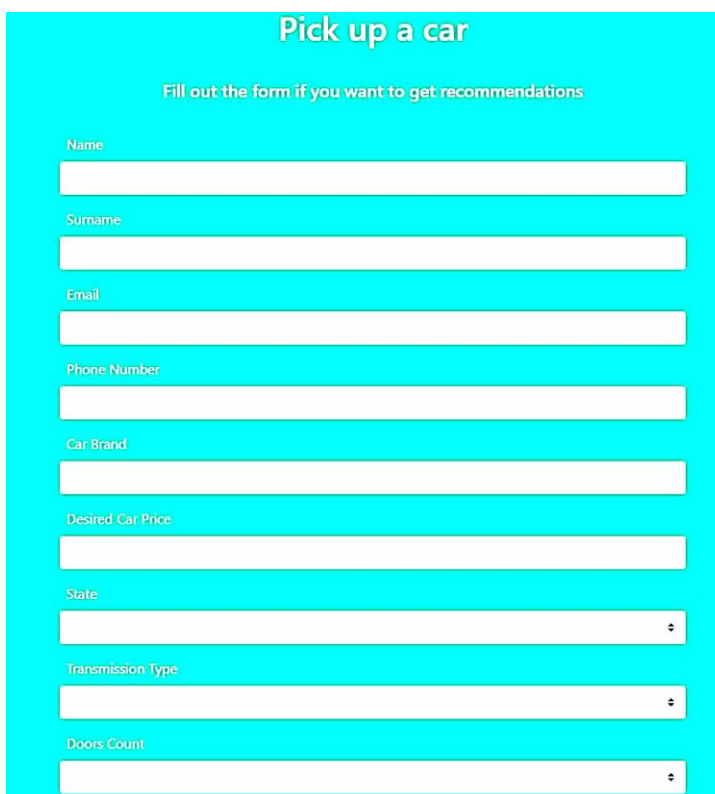
Якщо розглядати робітника компанії з продажу автомобілів, то він взагалі не працює з веб-застосунком, а лише з CRM системою. Під час роботи робітника

з CRM системою весь час задіяні лише два компоненти CRM система роботу з якою проводить робітник та CDS. Після кожної операції робітник компанії отримує повідомлення про помилку, якщо така відбулася, або результат виконання операції.

3.3. Інструкція користувача

3.3.1. Інструкція клієнта

Для отримання рекомендацій користувачу необхідно перейти до веб-застосунку та заповнити форму заявки на підбір автомобіля. На рисунку 3.7. зображено форму заявки на підбір автомобіля.

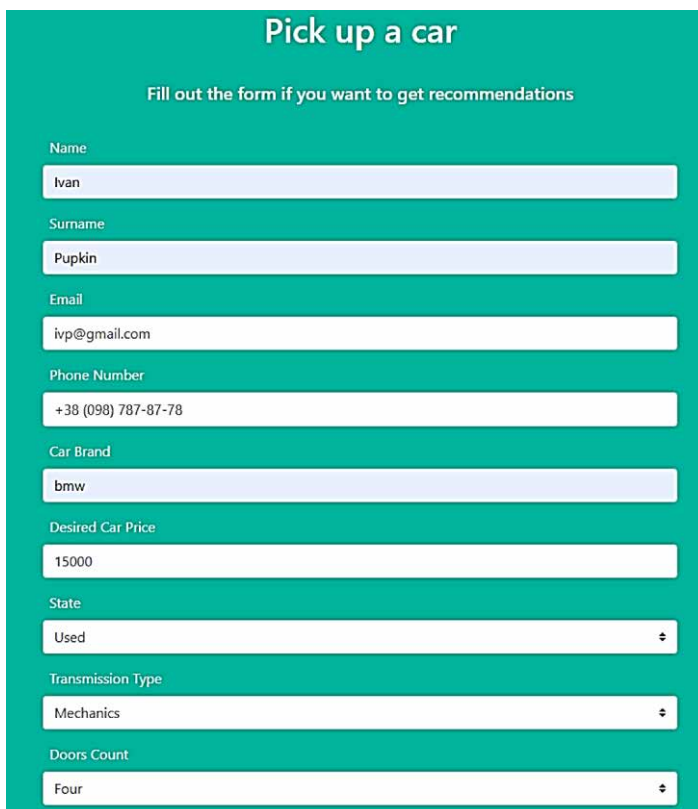


The image shows a web form titled "Pick up a car" with a subtitle "Fill out the form if you want to get recommendations". The form contains the following fields:

- Name:
- Surname:
- Email:
- Phone Number:
- Car Brand:
- Desired Car Price:
- State:
- Transmission Type:
- Doors Count:

Рис. 3.7. Форма заявки на підбір автомобіля

Після відкриття форми клієнт має заповнити персональні дані (ім'я, прізвище, email адресу), побажання по характеристикам автомобіля (марку автомобіля бажану ціну, кількість дверей, тип приводу, тип рушійної сили, обсяг споживання, стан, тип коробки передач) та натиснути на чекбокс тим самим підтвердивши згоду на опрацювання персональних даних. На рисунках 3.8. та 3.9. 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. 3.8. 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. 3.9. Приклад заповнення форми заявки на придбання автомобіля

Після заповнення форми клієнт має натиснути на кнопку «Sendrequest». Після успішної відправки запиту клієнт має побачити, що форма зникла та з'явилося повідомлення про те, що представник зв'яжеться з клієнтом. На

рисунку 3.10. зображено зміну сторінки та відображення повідомлення про те, що заявку прийнято.

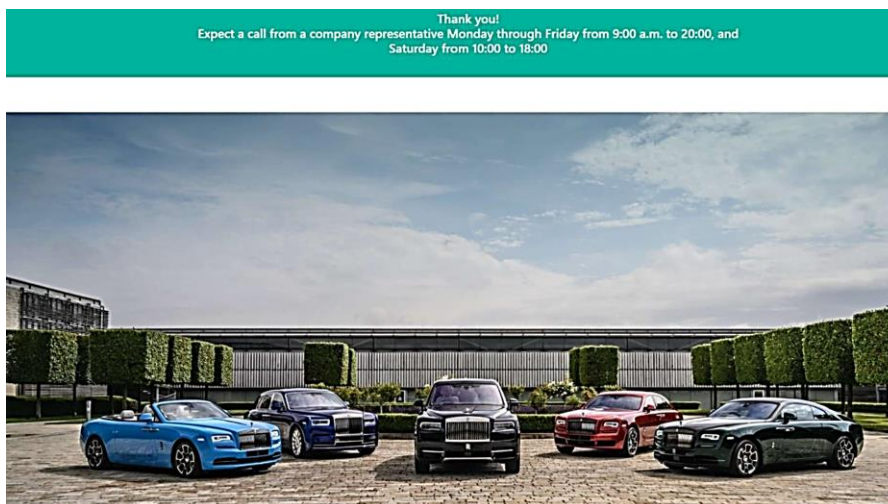


Рис. 3.10. Відображення повідомлення про прийняття заявки

3.3.2. Інструкція робітника компанії з продажу автомобілів

Для того, щоб розпочати роботи з CRM, робітник компанії з продажу автомобілів має авторизуватися. Для авторизації необхідно ввести логін та пароль та натиснути на кнопку «SignIn». На рисунках 3.11. та 3.12. зображено форми вводу логіна та паролю відповідно.

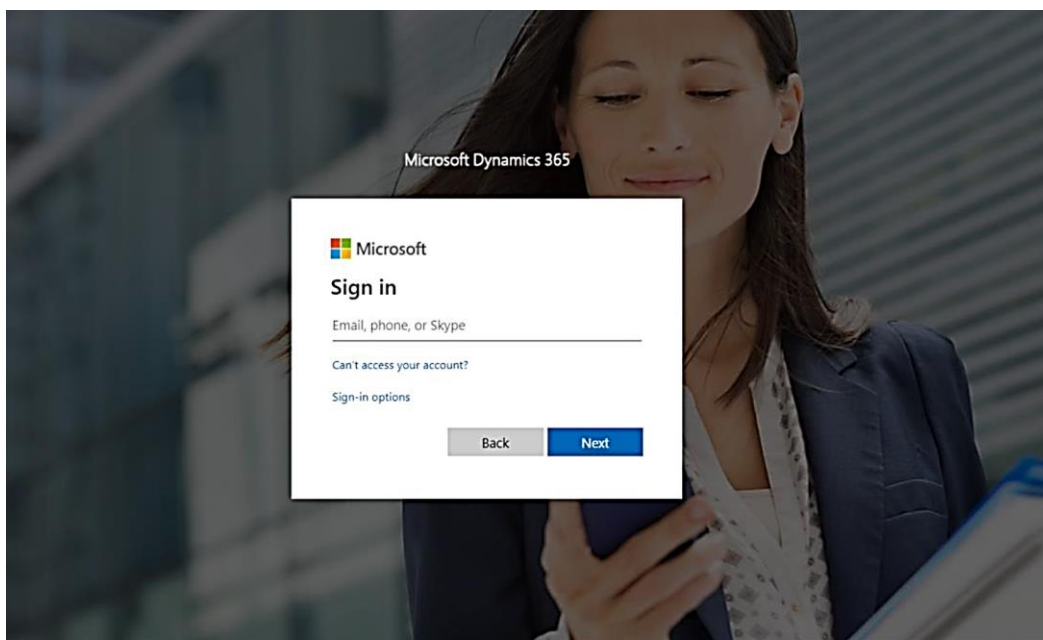


Рис. 3.1. Форма вводу логіна

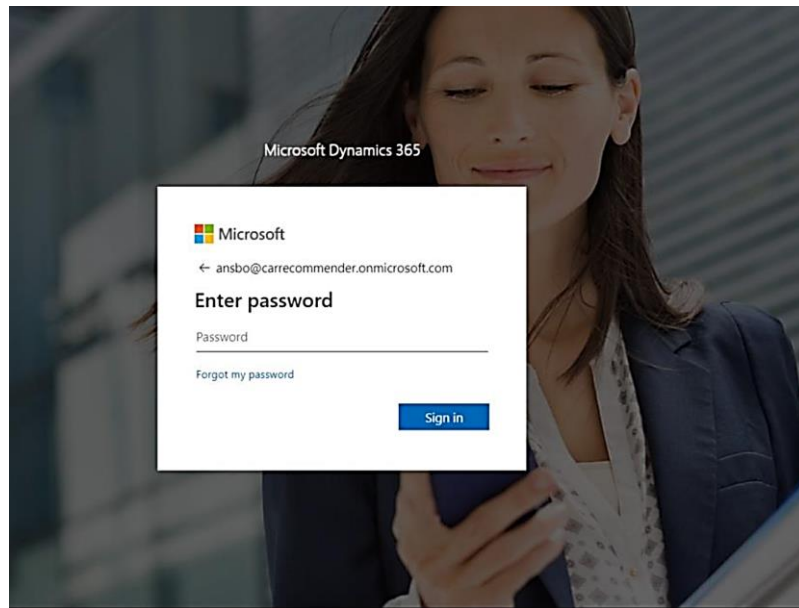


Рис. 3.12. Форма вводу паролю та входу в систему

Після авторизації робітник компанії потрапляє на головну сторінку CRM системи, де він може обрати необхідний пункт меню, перейти до налаштувань, переглянути діаграму кількості клієнтів, які потрапили з різних джерел. На рисунку 3.13. зображено головну сторінку CRM системи.

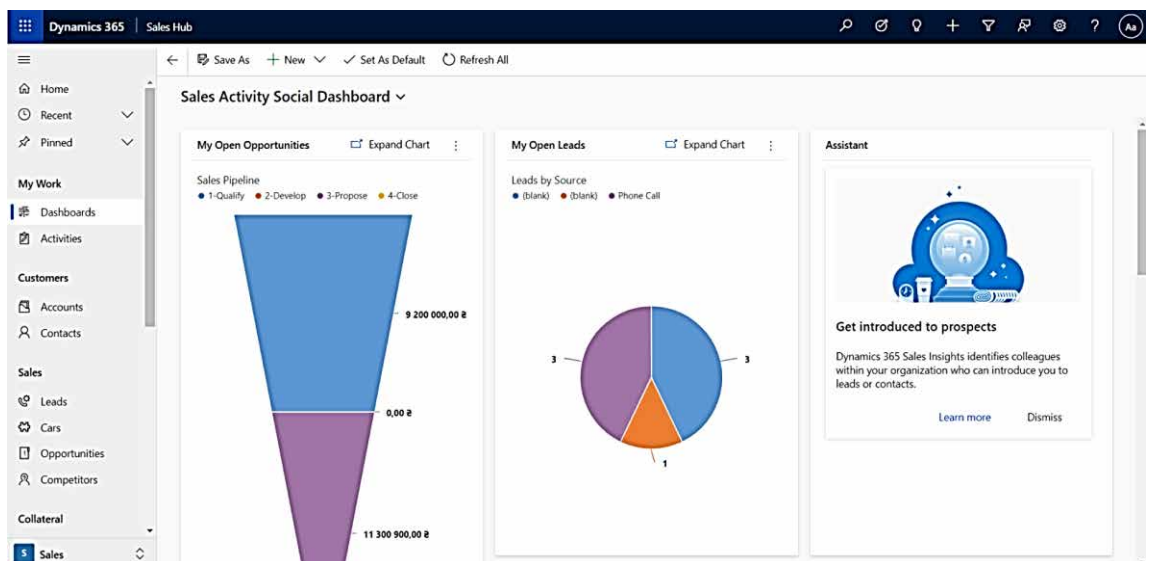


Рис. 3.13. Головна сторінка CRM системи

Якщо клієнт телефонує до компанії та хоче отримати рекомендації без використання веб-застосунку, то робітник компанії з продажу автомобілів має перейти до списку клієнтів натиснувши на «Leads» в меню зліва. Відображення списку клієнтів зображено на рисунку 3.14.

Name	Topic	Status Reason	Created On
Alex Wu	Expressed interest in A. Datum X line of print...	New	21.01.2017 0:52
Anton Sbooyk	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 Aberornbie	Likes our products	New	25.10.2020 3:22
Ylada Chernenko	Phone call 31/10/20 20:30	RecommendationsFound	31.10.2020 20:28
Yurii Sbooyk	Phone 31/10	CarSelected	31.10.2020 18:59

Рис. 3.14. Відображення списку клієнтів

Для того, щоб зареєструвати нове звернення робітник натискає на кнопку «New» з верхнього меню. Форма створення нового клієнта зображена на рисунку 3.15.

The screenshot shows the 'New Lead' form in Dynamics 365. The form is titled 'New Lead' and has a 'Lead Source' of 'Warm'. The 'Summary' tab is active, showing a 'Contact' section with the following fields: Topic, First Name, Last Name, Business Phone, Mobile Phone, Email, and Is Consent Given. The 'Timeline' section on the right shows a message 'Almost there' with the text 'Select Save to see your timeline.'.

Рис. 3.15. Форма створення нового клієнта

Робітник має заповнити персональні дані клієнта такі, як ім'я, прізвище, номер телефону, email, вказати тему звернення. Приклад заповнення форми персональними даними та вибір джерела звернення зображено на рисунку 3.16.

The screenshot shows the 'New Lead' form in Dynamics 365, now filled with personal data. The 'Lead Source' is 'Phone Call'. The 'Summary' tab is active, showing the 'Contact' section with the following fields: Topic (Phone Call 07/11/20), First Name (Petr), Last Name (Petrov), Business Phone, Mobile Phone (380577575542), Email (pp@ukr.net), and Is Consent Given (Yes). The 'Timeline' section on the right shows a message 'Almost there' with the text 'Select Save to see your timeline.'.

Рис. 3.16. Приклад заповнення форми персональними даними та вибору джерела звернення

Після заповнення персональних даних робітник має перейти до вкладки «CarSelection». Форма заповнення побажань клієнта та роботи з клієнтом по підбору автомобіля зображена на рисунку 3.17.

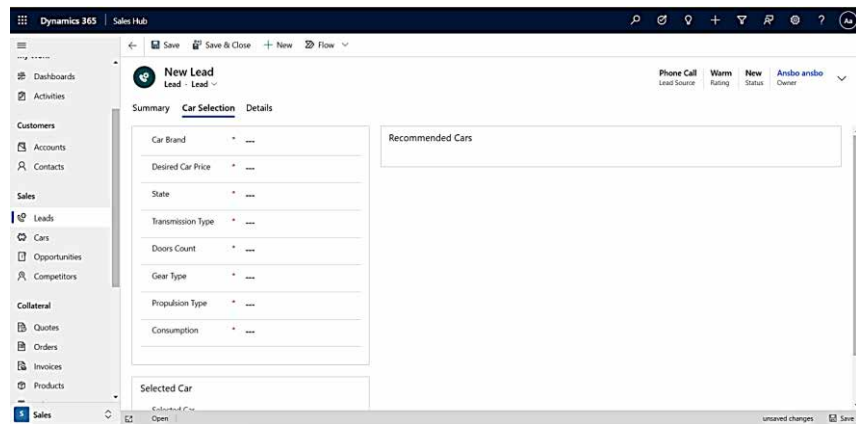


Рис. 3.17. Форма заповнення побажань клієнта та роботи з клієнтом по підбору автомобіля

Робітник має заповнити побажання клієнта щодо автомобіля, а саме марку автомобіля, споживання палива, бажану вартість, кількість дверей, тип двигуна, тип приводу, тип рушійної сили. Приклад заповнення форми побажань клієнта та роботи з клієнтом по підбору автомобіля зображена на рисунку 3.18.

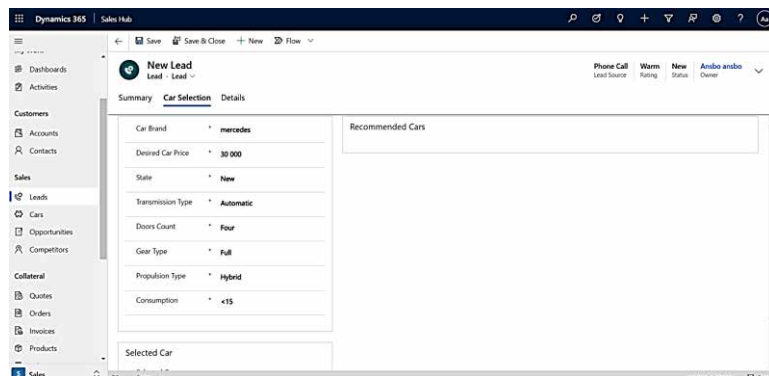


Рис. 3.18. Приклад заповнення форми побажань клієнта Розглянемо список дій, які має виконати робітник, якщо заявка від клієнта потрапляє з веб-застосунку. Робітник має обрати подання «AllLeads» та переглянути список клієнтів, що мають статус «New». Відображення списку всіх клієнтів зображено на рисунку 3.19.

Name	Topic	Owner	Status Reason	Created On
Petr Petrov	Phone call 07/11/20	Arbno arbno	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	Recommendations found	05.11.2020 20:48
Petr Petrov	Phone call 03/11/20 20:26	Arbno arbno	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
Masha Chernenko	Phone call 31/10/20 20:30	Arbno arbno	No Longer Interested	31.10.2020 20:28
Yuri Stoyrk	Phone 31/10	Arbno arbno	CarSelected	31.10.2020 18:59
Anton Stoyrk	Phone call 28/10	Arbno arbno	No Longer Interested	28.10.2020 20:13
George Sullivan	Store is expanding - send new literature	Arbno arbno	New	25.10.2020 3:22
Kim Abercrombie	Likes our products	Arbno arbno	New	25.10.2020 3:22
Adrian Dumitracu	Video Hardware Upgrade	David So (Sample Data)	CarBought	25.10.2020 3:22

Рис. 3.19. Відображення списку всіх клієнтів

Робітник має обрати клієнта зі статусом «New» та натиснути два рази лівою кнопкою миші на відповідний рядок. Після натискання відкривається форма клієнта з усіма даними, що клієнт заповнив в веб застосунку, а також джерело надходження звернення вказано, як «WebApp». На рисунках 3.20 та 3.21 зображено приклади відображення персональних даних та побажань клієнта, які клієнт заповнив в веб- застосунку.

Ivan Pupkin
Lead - Lead

Summary Car Selection Details Related

Contact
Topic*
Web app request 07-Nov-20 18:10:14

First Name*
Ivan

Last Name*
Pupkin

Business Phone

Mobile Phone
38067878778

Email
ip@sigmal.com

In Consent Given*
Yes

Timeline

Auto post on Ivan Pupkin
Integration user created Ivan Pupkin
18.10

Рис. 3.20. Відображення персональних даних клієнта

Ivan Pupkin
Lead - Lead

Summary Car Selection Details Related

Car Brand *
bmw

Desired Car Price *
15 000

State *
Used

Transmission Type *
Mechanics

Doors Count *
Four

Gear Type *
Back

Propulsion Type *
Fuel

Consumption *
415

Recommended Cars

No data available

Рис. 3.21. Відображення побажань клієнта

Процес по роботі з клієнтом є однаковим для звернень, які надходять з веб- застосунку та в режимі телефонного дзвінка. Розглянемо даний процес для клієнта звернення, якого прийшло з веб застосунку. Для початку роботи з

клієнтом робітник має змінити статус запису з «New» на «StartedWork». Зміна статусу запису та збереження запису зображена на рисунках 3.22 та 3.23.

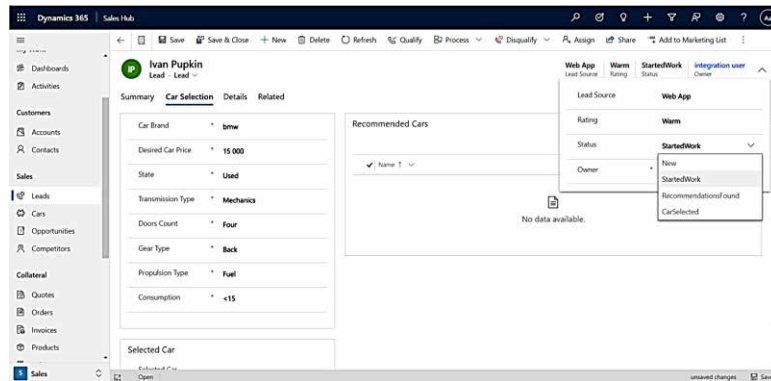


Рис. 3.22. Зміна статусу запису

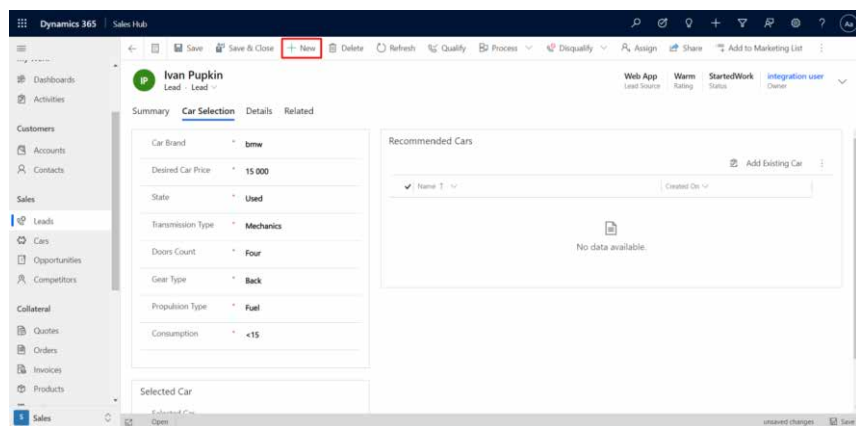


Рис. 3.23. Збереження запису

Після зміни статусу запису на «StartedWork» система знаходить рекомендовані автомобілі, відображає їх, як пов'язані записи на формі в таблиці RecommendedCars. Також система автоматично змінює статус запису на «RecommendationsFound». Відображення списку рекомендацій зображено на рисунку 3.24.

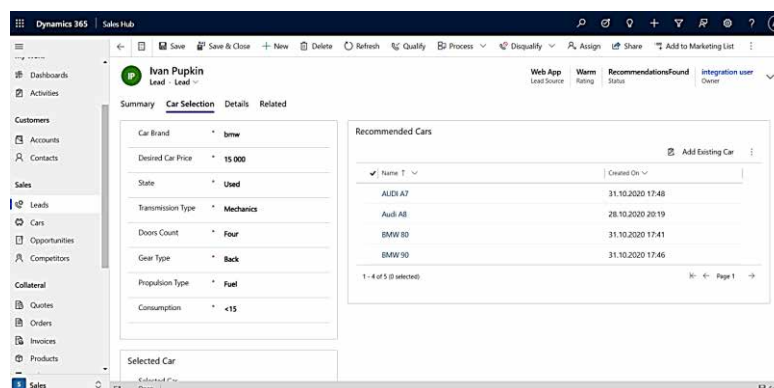


Рис. 3.24. Відображення списку рекомендацій

Після знаходження рекомендацій робітник компанії з продажу

автомобілів проводить комунікацію з клієнтом за допомогою мобільного телефону або email, які вказав клієнт при подачі заявки на знаходження рекомендацій. Після того, як клієнт обирає автомобіль з списку рекомендацій, робітник вказує посилання на обраний автомобіль в полі «SelectedCar» та зберігає запис. Після заповнення даного поля обраний автомобіль деактивується (при підборі автомобілів для інших клієнтів даний автомобіль не буде враховуватись) та зникає з таблиці рекомендацій. Вибір автомобіля та оновлення таблиці рекомендацій зображено на рисунку 3.25.

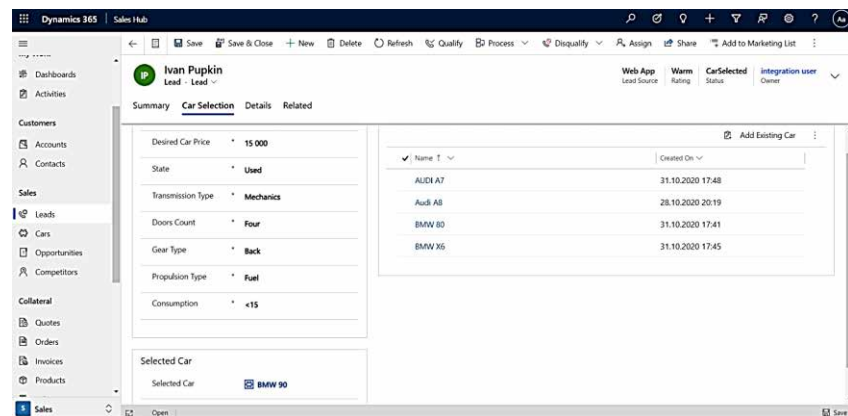


Рис. 3.25. Вибір автомобіля та оновлення таблиці рекомендацій

Якщо клієнт обирає інший автомобіль, або на даний момент змінив думку на рахунок вибору автомобіля, то робітник компанії з продажу автомобілів має очистити поле «SelectedCar» та зберегти запис. Автомобіль, що був обраний раніше та деактивований буде активований, з'явиться в таблиці рекомендацій, та буде доступний для знаходження рекомендацій для інших клієнтів. Результат очищення поля вибраного автомобіля зображено на рисунку 3.26.

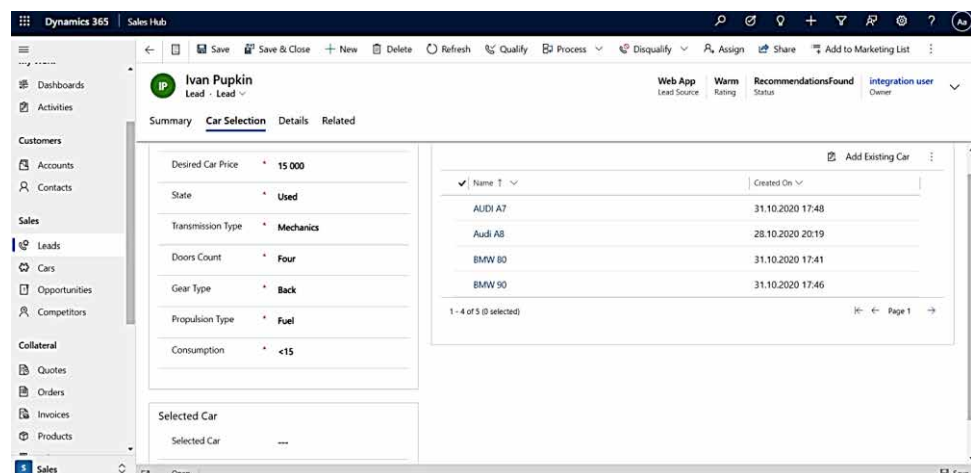


Рис. 3.26. Результат очищення поля вибраного автомобіля

Якщо клієнт відмовляється від покупки обраного автомобіля, робітники не можуть зв'язатися з клієнтом за вказаними контактними даними, то робітник має змінити статус клієнта на відповідний натиснувши на кнопку «Disqualify». При такій зміні статусу обраний клієнтом автомобіль стає активним та в подальшому враховується в знаходженні рекомендацій для інших клієнтів. Процес зміни статусу клієнта зображено на рисунку 3.27.

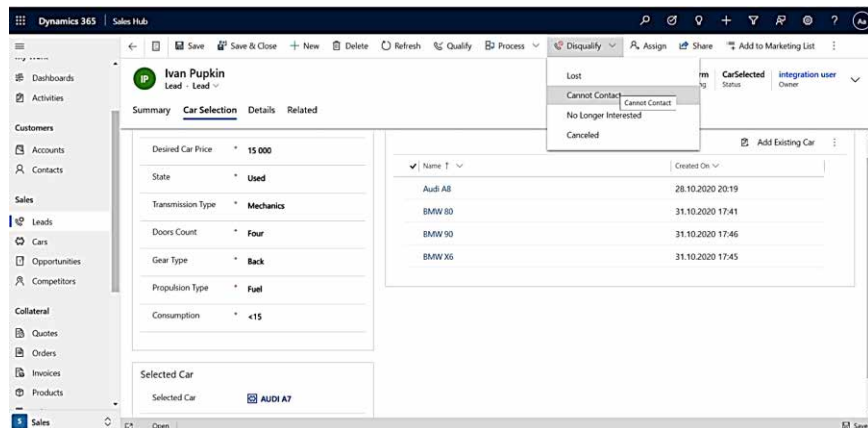


Рис. 3.27. Процес зміни статусу клієнта

Якщо клієнт придбав автомобіль, то робітник має змінити статус клієнта на «CarBought» натиснувши на кнопку «Qualify». Обраний автомобіль стане неактивним та в подальшому не буде використовуватися при підборі автомобілів. Результат зміни статусу клієнта на «CarBought» зображено на рисунку 3.28.

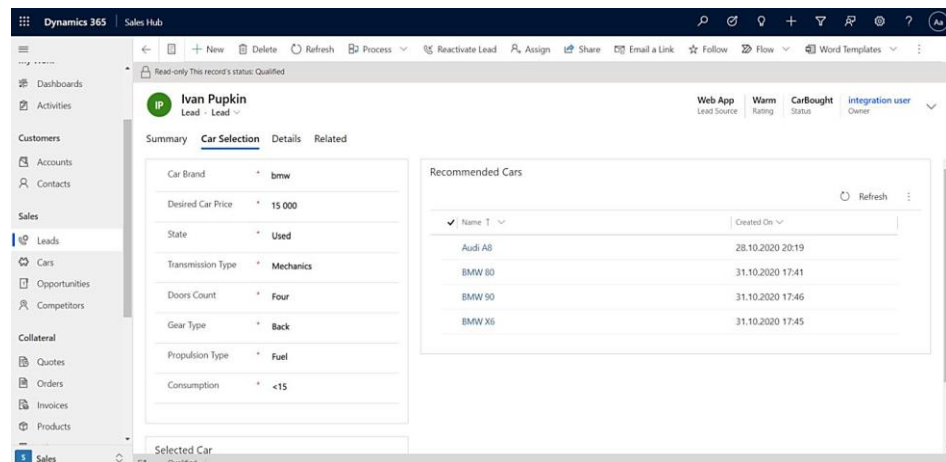


Рис. 3.28. Процес зміни статусу клієнта

Розглянемо процес управління списком автомобілів. Для перегляду списку доступних автомобілів в системі треба натиснути на кнопку «Cars» в

меню зліва. Перехід до списку автомобілів та відображення списку автомобілів зображено на рисунках 3.29. та 3.30. відповідно.

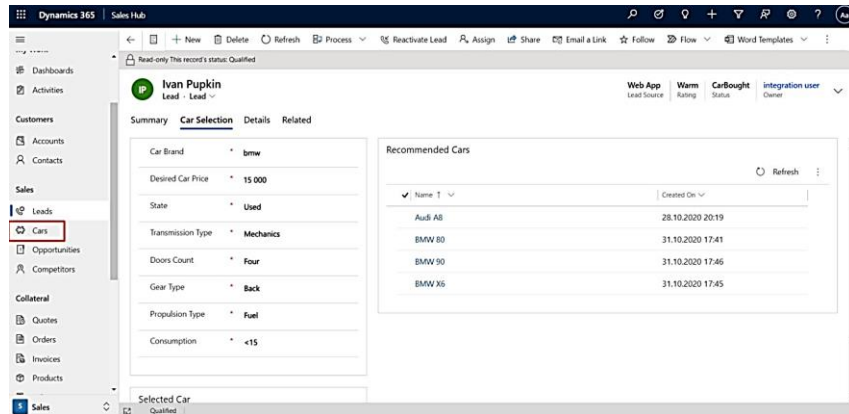


Рис. 3.29. Перехід до списку автомобілів

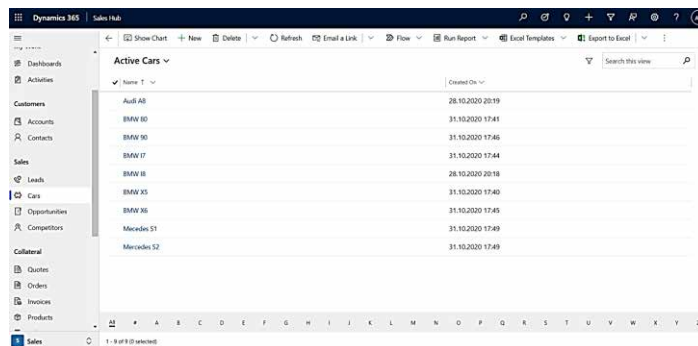


Рис. 3.30. Відображення списку автомобілів

Для того, щоб створити автомобіль в системі робітник натискає на кнопку «New» з верхнього меню. Форма створення нового автомобіля зображена на рисунку 3.31.

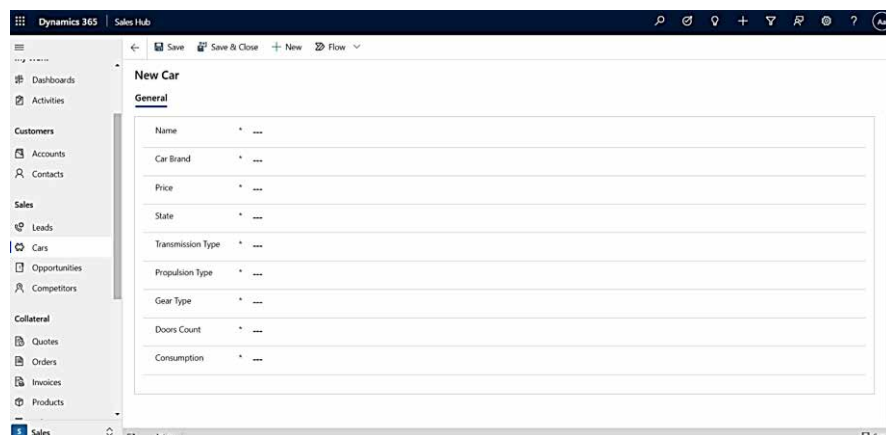


Рис. 3.31. Форма створення нового автомобіля

Робітник компанії з продажу автомобілів заповнює параметри автомобіля та натискає на кнопку «Save». Приклад заповнення форми створення автомобіля зображено на рисунку 3.32.

Рис. 3.32. Приклад заповнення форми створення автомобіля
Результат створення автомобіля зображено на рисунку 3.33.

Рис. 3.33. 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».

3.4. Опис технічного забезпечення

При розробці веб-застосунку для збереження звернень клієнтів було використано клієнт-серверну архітектуру, також сама CRM система побудована за клієнт-серверною архітектурою. Діаграма розгортання показує компоненти, що використовуються в системі, та способи зв'язку компонентів між собою. Структурну схему розгортання наведено в графічному матеріалі. Перевагою розробленої архітектури є те, що замість веб-застосунку може використовуватися будь-яке джерело даних, наприклад: чат бот, інша система.

Також одночасно в рекомендаційну систему заявки можуть відправлятися з декількох джерел.

3.5. Висновки до розділу 3

В розділі опису програмного та технічного забезпечення було описано технології та засоби, що використовувались під час розробки рекомендаційної системи підбору автомобілів для продажу клієнтам, розроблено структурні схеми класів, структурну схему послідовності, структурну схему розгортання, інструкцію користувача.

РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1. Опис ідеї проекту

Опис ідеї проекту описує зміст проекту, напрямки застосування проекту, вигоди для користувача. Опис ідеї проекту наведено в таблиці 4.1.

Таблиця 4.1.

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
1	2	3
Рекомендаційна система підбору автомобілів для продажу клієнтам. Для заповнення заявок на підбір автомобілів 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.	Використання рекомендаційної системи для знаходження рекомендацій для клієнтів після отримання звернень.	<ul style="list-style-type: none"> ➤ зменшення витрат часу робітників компанії на знаходження рекомендацій; ➤ збільшення кількості оброблених звернень за певний період часу.
	Використання рекомендаційної системи для управління процесом роботи з клієнтом.	<ul style="list-style-type: none"> ➤ збільшення точності відслідковування статусу обробки заявки від клієнта; ➤ забезпечення існування єдиної бази звернень;
	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.

Для того, щоб зрозуміти чи є мій проект конкурентоспроможним

визначаються слабкі, нейтральні та сильні сторони в порівнянні з проектами конкурентів. Визначення сильних, слабких та нейтральних характеристик ідеї проекту наведено в таблиці 4.2.

Таблиця 4.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні)товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	carandbike	nadaguides	infocar			
1	2	3	4	5	6	7	8	9
1	Легкість реалізації	Легко реалізувати	Вже реалізований	Вже реалізований	Вже реалізований		+	
2	Знаходження рекомендацій	Рекомендаційна система знаходить рекомендації, але не повертає всі автомобілі профільтовані по обраними клієнтом параметрами.	Знаходить рекомендації	Повертає всі автомобілі профільтовані по обраними клієнтом параметрами.	Повертає всі автомобілі профільтовані по обраними 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	+		
5	Зручний інтерфейс по управлінню даними клієнтів, автомобілів	Інтерфейс зручний 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 рахунок існування Web API	Заявки зотримуються 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		+	
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 роботи з даними автомобілів, клієнтів та управління процесом роботи з клієнтом	Легко подається нова бізнес логіка	Дані не знайдено	Дані не знайдено	Дані не знайдено			+
11	Масштабованість системи	При збільшенні кількості наявних автомобілів системі процес пошуку рекомендацій відбувається швидко	Дані не знайдено	Дані не знайдено	Дані не знайдено			+
12	Можливість використання даних клієнтів маркетингових цілях	Легко можна інтегрувати маркетинговий	Відсутня	Відсутня	Відсутня			+

4.2. Технологічний аудит ідеї проекту

Проведемо технологічний аудит технологій, які необхідні для реалізації проекту. Перелік технологій, які будуть використовуватись під час реалізації проекту, аналіз наявності та доступності необхідних технологій наведено в таблиці 4.3.

Таблиця 4.3.

Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	2	3	4	5
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 реалізації проекту – можлива. Для знаходження рекомендацій було обрано алгоритм знаходження рекомендацій на основі побажань клієнта. Для зберігання та ведення даних автомобілів, клієнтів, звернень та управління процесом по роботі з клієнтом було обрано CRM систему Microsoft Dynamics 365 forSales. Для розробки Web API було обрано фреймворк NetCore 2.2, що використовує мову C#. Для розробки веб застосунку для заповнення заявок на знаходження рекомендованих автомобілів було обрано фреймворк Bootstrap та бібліотеку JQuery, яка побудована на мові програмування JavaScript.

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Для планування напряму розвитку проекту спочатку необхідно проаналізувати попит на ринку. Аналіз попиту наведено в таблиці 4.4.

Таблиця 4.4.

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж	
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Наявні такі обмеження: 1. необхідність дослідження запропонованого алгоритму для знаходження рекомендацій на великому обсязі наявних даних; 2. необхідність дослідження поведінки системи при можливих неполадках системи
5	Специфічні вимоги до стандартизації та сертифікації	

По результатах аналізу попиту можна вважати, що ринок є привабливим для входження.

Після аналізу попиту сформуємо потенційні групи клієнтів стартап-проекту та визначимо вимоги до проекту. Характеристики потенційних клієнтів стартап- проекту наведено в таблиці 4.5.

Таблиця 4.5.

Характеристика потенційних клієнтів стартап-проекту

п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності поведінці потенційних цільових клієнтів	Вимоги споживачів до товару
	Потреба рекомендаційній системі, яка зменшить час знаходження рекомендацій і тим самим збільшить обсяги оброблених заявок.	Автомобільні салони; Компанії з продажу автомобілів; Особи підприємці, які продають автомобілі.	Всі потенційні цільові групи клієнтів потребують рекомендаційну систему, яка буде містити дані про автомобілі, клієнтів, заявки клієнтів та буде знаходити рекомендації на основі побажань клієнта з заявки.	1. Виконана з забезпеченням безпеки даних в системі; 2. Підтримка проекту та подальша співпраця для розширення функціонала системи за необхідністю; 3. Стабільність роботи системи.

Було сформовано три групи потенційних клієнтів стартап-проекту: автомобільні салони, компанії з продажу автомобілів, особи підприємці, які продають автомобілі. Наступним етапом аналізу ринкових можливостей стартап- проекту є аналіз ринкового середовища. В рамках даного аналізу визначаються фактори, що сприяють та заважають впровадженню проекту. Фактори загроз наведено в таблиці 4.6., а фактори можливостей наведено в таблиці 4.7.

Таблиця 4.6.

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Технології	Технології з плином часу застарівають та потребують оновлення.	Необхідно проводити перегляд програмних рішень раз в певний період часу та оновлювати використані технології

			для забезпечення стабільної та найбільш продуктивної роботи системи
2	Обсяг даних	Через певний час обсяг даних може стати занадто великим та досягти ліміту по обсягу даних в CRM системі	Запланувати періодичні роботи по аналізу та очищенню непотрібних даних.
3	Оновлення CRM системи	CRM система періодично оновлюється і можуть з'явитися оновлення, що будуть потребувати модернізації системи для коректної роботи.	Відслідковувати оновлення і при оновленнях, що потребують додаткової розробки та модернізації розробленого функціоналу, проводити оновлення впроваджених систем.

Таблиця 4.7.

Фактори можливості

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Можливість модернізації системи	Рекомендаційна система, може бути модернізована під потреби кожного клієнта та дозволяє впровадження автоматизації інших бізнес процесів за потребою клієнта	Впровадження системи з внесенням максимального задоволення потребам клієнта

2	Можливість інтеграції з різними джерелами вхідних заявок	Поступове збільшення джерел заявок клієнтів надання придбання автомобілів	Тестування інтеграцій з різними джерелами даних, розробка інтеграцій за потребою клієнта.
3	Можливість інтеграції з існуючими джерелами даних	За побажанням клієнта система може отримувати з існуючих систем зберігання роботи з даними.	Комунікація з розробниками систем, які наразі використовують клієнти. Забезпечення правильної інтеграції між системами.

Після визначення факторів загроз та факторів можливостей проведено аналіз пропозицій та визначимо загальні риси конкуренції на ринку. Ступеневий аналіз конкуренції на ринку наведено в таблиці 4.8.

Таблиця 4.8.

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	2	3
1. Тип конкуренції - чиста	На ринку існують системи, що мають схожий функціонал, але не є повними аналогами запропонованої рекомендаційної системи	Розвивати рекомендаційну систему за для легшого впровадження та більшого спектру функціоналу, допрацювати алгоритм для більшої точності рекомендацій.
2. За рівнем конкурентної боротьби - світова	На даний момент щодня обсяги продажу та купівлі автомобілів зростають.	Для поширення на світовому ринку перш за все необхідно отримати достатньо клієнтів

	Тому запропонована на локальному ринку та система буде необхідною зарекомендувати себе. в всіх країнах світу.	
3. За галузевою ознакою - внутрішньогалузева	Тип конкуренції для рекомендаційної системи є внутрішньогалузевим, тому, що рекомендаційна система використовується в сфері продаж автомобільної галузі.	Спланувати вдосконалення системи для того, щоб розширити спектр товарів, які можуть бути знайдені у якості рекомендацій.
4. Конкуренція за видами товарів: товарно-родова	Товарно-видова конкуренція між іншими системами, що мають подібний функціонал	Необхідно розвивати систему та надавати якісні послуги для того, щоб зарекомендувати себе на ринку
5. За характером конкурентних переваг - нецінова	Переважно вибір системи здійснюється не за ціною, а за спектром функціоналу та якістю надання послуг	Розробити коробочне рішення для швидкого впровадження і мінімального налаштування під клієнта.
6. За інтенсивністю - не марочна	При виборі системи дивляться здебільшого на функціонал, а не на назву компанії, якою маркується система	Зарекомендувати себе на ринку, показати клієнту на попередній демонстрації можливості системи та розповісти про можливість індивідуального налаштування.

Проведемо аналіз конкуренції в галузі за моделлю 5 сил М.Портера та визначимо перелік факторів конкурентоспроможності. Аналіз конкуренції в

галузі за М. Портером наведено в таблиці 4.9. Обґрунтування факторів конкурентоспроможності наведено в таблиці 4.10.

Таблиця 4.9.

Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	1	2	3	4	5
Складові аналізу	carandbike nadaguides infocar	Існує досить багато систем, що на основі побажань клієнта фільтрують список наявних автомобілів та відображають його. Досить рідко зустрічаються системи, що насправді надають рекомендації	Запропонована рекомендаційна система не потребує постачальників	Факторами сили клієнтів є контроль стабільності та швидкості роботи системи, а також точність знаходження рекомендацій	Товари замінники не пропонують аналогічного функціоналу, який надає запропонована рекомендаційна система
Висновки:	Конкурент на боротьба на ринку з боку прямих конкурентів не є значною	Вихід на ринок є можливим та згідний момент відсутні потенційні конкуренти	Постачальники взагалі не впливають на роботу ринку	Клієнти диктують умови роботи ринку тому, що основною ціллю впровадження системи є задоволення потреб клієнта.	Товарів-замінників не було знайдено.

Таблиця 4.10.

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	2	3
1	Знаходження рекомендацій на основі побажань клієнта	Алгоритм знаходить рекомендації для клієнта на основі його побажань і не потребує попередньої історії дій для даного клієнта.

2	Збереження запитів на придбання автомобілів в системі	Рекомендаційна система зберігає всі запити та в подальшому дозволяє проводити аналіз отриманих даних
3	Легка інтеграція з іншими системами	Рекомендаційна система використовує Web API і тому легко налаштувати інтеграцію з іншими системами.
4	Відсутність потреби у розміщенні в конкретному місці	Компоненти системи не потребують конкретного типу розміщення і тому можуть бути розміщені за побажанням клієнта на серверах або в хмарі.
5	Легка модернізованість системи під потреби клієнта	Використання CRM системи дає можливість легко модернізувати функціонал системи під потреби конкретного клієнта.

Проведемо аналіз сильних та слабких сторін стартап-проекту. Та складемо SWOT-аналіз. Порівняльний аналіз сильних та слабких сторін наведено в таблиці 4.11. SWOT-аналіз наведено в таблиці 4.12.

Таблиця 4.11.

Порівняльний аналіз сильних та слабких сторін

п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з рекомендаційною системою						
			-3	-2	-1	0	+1	+2	+3
1	2	3	4	5	6	7	8	9	10
1	Знаходження рекомендацій на основі побажань клієнта	19		+					

2	Збереження запитів на придбання автомобілів в системі	20	+						
3	Легка інтеграція з іншими системами	18		+					
4	Відсутність потреби у розміщенні в конкретному місці	15				+			
5	Легка модернізованість системи під потреби клієнта	15				+			

Таблиця 4.12.

SWOT- аналіз стартап-проекту

Сильні сторони	Слабкі сторони
<p>Знаходження рекомендацій на основі побажань клієнта</p> <p>Збереження запитів на придбання автомобілів в системі</p> <p>Зменшення заторів у мережі</p> <p>Легка інтеграція з іншими системами</p> <p>Відсутність потреби у розміщенні в конкретному місці</p> <p>Легка модернізованість системи під потреби клієнта</p>	<p>Відсутність моментального відображення рекомендацій клієнту</p>
Можливості	Загрози

Можливість модернізації системи	Обсяг даних
Можливість інтеграції з різними джерелами вхідних заявок	Оновлення CRM системи
Можливість інтеграції з існуючими джерелами даних	Технології

Проаналізуємо альтернативи ринкової поведінки для виведення проекту на ринок та початку проваджень систем клієнтам. Альтернативи ринкового впровадження стартап-проекту наведено в таблиці 4.13.

Таблиця 4.13.

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Впровадження системи для бажаючих клієнтів на ринку, отримання відгуків про систему, виправлення проблем	40%	3 місяці
2	Розгортання системи на середовищі розробки, тестування системи, відлагодження системи, тестування системи фокус групами, фінальне налаштування системи, виведення на ринок.	75%	5,5 місяців

Було обрано другу альтернативу ринкової поведінки – тестування системи на середовищі розробки, фіналізація розробленого рішення і лише потім виведення системи на ринок.

4.4. Розробка ринкової стратегії проекту

Визначимо стратегію охоплення ринку та базову стратегію розвитку. Вибір цільових груп потенційних споживачів наведено в таблиці 4.14, визначення базової стратегію розвитку – в таблиці 4.15.

Таблиця 4.14.

Вибір цільових груп 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 3 групи, як цільові тому, що потреби кожної цільової групи не відрізняються. Відмінність між ними лише в обсягу продаж та кількості наявних даних. Завдяки вибору 3-х цільових груп буде досягнуто більший охопит ринку і швидше зростання репутації.

Отже, будемо використовувати стратегію масового маркетингу.

Таблиця 4.15.

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	2	3	4

Розгортання системи на середовищі розробки, тестування системи, відлагодження системи, тестування системи фокус групами, фінальне налаштування системи, виведення на ринок.	Стратегія масового маркетингу	Стабільність пропонованої системи, відсутність помилок, що будуть заважати роботі системи.	Стратегія спеціалізації
---	-------------------------------	--	-------------------------

Визначення базової стратегії конкурентної поведінки наведено в таблиці 4.16.

Таблиця 4.16.

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» 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

Визначення стратегії позиціонування наведено в таблиці 4.17.

Таблиця 4.17.

Визначення 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. Високий рівень підтримки після впровадження.
2	Підтримка проекту та подальша співпраця для розширення функціонала системи за необхідністю.	Стратегія спеціалізації		Готовність модернізувати систему під потреби клієнта. Стабільна робота Системи.
3	Стабільність роботи системи.	Стратегія спеціалізації		

4.5. Розроблення маркетингової програми стартап-проекту

На основі результатів конкурентоспроможності сформуємо маркетингові концепції товару. Визначення ключових переваг концепції потенційного товару наведено в таблиці 4.18.

Таблиця 4.18.

Визначення 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 тим самим збільшить обсяги оброблених заявок.	Зменшення витрат часу працівників 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 моделі товару наведено в таблиці 4.19.

Таблиця 4.19.

Опис трьох рівнів моделі 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	Властивості/характеристики	/Нм	Вр/Тх/Тл/ Е/Ор

виконанні	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 патентуванням алгоритму та принципу роботи системи.

Далі сформуємо систему збуту та концепцію маркетингових комунікацій. Формування системи збуту наведено в таблиці 4.20. Концепція маркетингових комунікацій наведена в таблиці 4.21.

Таблиця 4.20.

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глиби на каналу збуту	Оптимальна система збуту

1	Підписання контракту на впровадження і підтримку системи	Впровадження системи на середовищі замовника	Однорівнева	Власна система збуту
---	--	--	-------------	----------------------

Таблиця 4.21.

Концепція 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 рекомендаційної системи та зацікавити до впровадження	Розробка маркетингових повідомлень, описів для маркетингових посилань, розробка сайту компанії з описом функціоналу рекомендаційної системи

4.6. Висновки до розділу 4

У розділі розробки стартап-проекту було виконано опис ідеї, проведено технологічний аудит ідеї проекту, проаналізовано ринкові можливості запуску стартап-проекту, визначено фактори конкурентоспроможності, розроблено ринкову та маркетингові стратегії.

В результаті проведеного аналізу можна зробити висновки, що рекомендаційна система повинна мати попит на ринку, тому що на ринку не існує систем з аналогічним функціоналом. Було обрано таку альтернативу ринкової

поведінки: розгортання системи на середовищі розробки, тестування системи, відлагодження системи, тестування системи фокус групами, фінальне налаштування системи, виведення на ринок. Дана альтернатива забезпечить, що при виході системи на ринок вона буде працювати стабільно і без системних помилок. Тому слід вважати, що подальше розроблення системи є доцільним.

ВИСНОВКИ

У результаті роботи над магістерською кваліфікаційною роботою було досліджено існуючий бізнес-процес пошуку рекомендацій за бажаними характеристиками для клієнта компанії, визначено, що пошук виконується менеджером власноруч, також проаналізувавши наявні методи та алгоритми пошуку рекомендацій виявлено, що вони не дають змоги знайти рекомендації, якщо клієнт не має історії в компанії. Розроблено концептуальну модель процесу автоматизованого пошуку рекомендацій з використанням методології IDEF0 для клієнтів, які не мають історії в компанії.

Створено алгоритм пошуку рекомендацій клієнтам, що не мають історії в компанії. Алгоритм базується на методах фільтрації вмісту та пошуку лексикографічної відповідності між вектором бажаних характеристик та векторами характеристик автомобілів, які є в компанії.

Експериментально було досліджено, що час роботи алгоритму залежить від збільшення кількості векторів характеристик автомобілів в компанії та при збільшенні кількості в 25 разів час збільшився в 5 разів, що для 2500 тисяч векторів характеристик сягає менше однієї секунди. Також, середнє значення коефіцієнта відповідності зменшується зі збільшенням загальної кількості векторів.

Експерименти показали, що алгоритм є стійким до збільшення загальної кількості векторів характеристик автомобілів в системі.

В результаті досліджень розроблено архітектуру рекомендаційної системи, яка складається з клієнтської та серверної частини веб застосунку, CRM системи на платформі Dynamics 365, CDS сервісу та наведено переваги розробленої архітектури.

На основі розробленої архітектури було реалізовано рекомендаційну систему на платформі Dynamics 365, що використовує розроблений алгоритм.

В результаті розробки стартап-проекту можна зробити висновки, що рекомендаційна система повинна мати попит на ринку, тому що на ринку не існує

систем з аналогічним функціоналом. Обрана альтернатива ринкової поведінки при виході системи на ринок забезпечить стабільність її роботи. Тому слід вважати, що подальше розроблення системи є доцільним.

Результатом магістерської роботи є дослідження, які привели до створення алгоритму пошуку рекомендацій для продажу автомобіля клієнтам, які не мають історії в компанії. Розроблена рекомендаційна система реалізує створений алгоритм, яка дозволить досягнути мету дослідження – підвищення ефективності роботи менеджерів, що в цілому збільшить прибуток компанії за рахунок збільшення загальної кількості оброблених заявок і продаж.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Л. Г.Ковалевський «Світовий автомобільний ринок: сучасний стан, особливості та перспективи розвитку» // Зовнішня торгівля: економіка, фінанси, право. - 2015. - № 5-6.- С. 60–67. - Режим доступу:
2. А. Ю.Сборик, Т. О.Телишева «АВТОМАТИЗОВАНА РЕКОМЕНДАЦІЙНА СИСТЕМА ПІДБОРУ АВТОМОБІЛІВ ДЛЯ ПРОДАЖУ КЛІЄНТАМ».
3. А.Ю.Сборик «Рекомендаційна система підбору автомобілів для продажу клієнтам»/ А.Ю. Сборик, Т.О. Телишева // IV ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ МОЛОДИХ ВЧЕНИХ ТА СТУДЕНТІВ «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ УПРАВЛІННЯ» НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020 р.-С.115-118.
4. Рівень автомобілізації в Україні в 2019 році. URL: <https://ecologia.com/news/nova-norma-do-yakoui-maye-pragnuty-ukrayina-mobilnist-z-nulovumu-vukydamy-dlya-vsih> (дата звернення: 04.03.2021).
5. Автомобільна промисловість / Вікіпедія – Україна. URL: http://uk.wikipedia.org/wiki/Автомобільна_промисловість (дата звернення: 05.03.2021).
6. TOAD Pro: Perfect OBD2 Programming Software For Laptop. URL: <https://www.obdadvisor.com/toad-pro/> (дата звернення: 10.03.2021).
7. Autoenginuity ScanToolURL: <https://www.autoenginuity.com/scantool/> (дата звернення: 20.03.2021).
8. PCMSCAN Features Palmer performance engineering, inc. URL: <https://www.palmerperformance.com/products/pcmscan/index.php> (дата звернення: 30.03.2021).
9. ProScan All-In-One Computer Aided Scanning Program URL: <https://www.proscan.org/> (дата звернення: 01.04.2021).
10. Мастакар, Гаурав. "Експериментальний аналіз безпеки сучасного автомобіля" . Університет Вашингтона та Каліфорнійський університет Сан-Дієго.

11. Marks, Paul . "Гаджет на 25 доларів дозволяє хакерам захопити контроль над автомобілем" . Новий вчений. 2020.
12. Van den Brink, Rob (July 10, 2012). "Dude, Your Car is Pwnd" (PDF). SANS Institute. Archived from the original (PDF) on February 23, 2015.
13. E / E Diagnostic Test Modes - Equivalent to ISO / DIS 15031-5: April 30, 2002 URL: <https://www.sae.org/standards/> (дата звернення: 11.04.2021).
14. Голян В.В., Кравченко О.К. Порівняння моделей життєвих циклів програмного забезпечення з метою виявлення найефективнішого// Збірник наукових праць ХНУ ПС2019р. – 6 с.
15. Viktoriia Skovorodnikova, Vladyslav Apukhtin, Mariya Shirokopetleva. The Relevance of Using Message Brokers in Robust Enterprise Applications Тези доповіді 2019 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2019 - pp. 305-309
16. Natalia Kravets, Tseshkovskyi N., Liutova K. S. Containers and virtual machines in microsoft Azure // Science, society, education: topical issues and development prospects. Abstracts of the 7th International scientific and practical conference. SPC "Sci-conf.com.ua". Kharkiv, Ukraine. 2020. P. 278-281. ЗАЮЗАТЬ
17. Мова програмування С# та .NET / Metanit. URL: <https://metanit.com/shart/general.php> (дата звернення: 15.04.2021).
18. World's largest developer community / Найбільша спільнота розробників у світі / Stackoverflow. URL: <https://stackoverflow.com/> (дата звернення: 19.04.2021).
19. Шилдт, Г. С# 4.0 Полное руководство / Г. Шилдт. – М.: "Вильямс", 2017.– 1056 с.
20. Основи систем баз даних / Дж. Д. Ульман - М. : Фінанси і статистика, 2023. – 334 с.
21. Уроки по С# і платформа .NET / Уроки по С# та платформі .NET / ProfessorWeb. URL: <http://professorweb.ru/> - 04.06.2019 г. (дата звернення: 28.04.2021).
22. Protocol-Oriented Programming [Електронний ресурс] // Хабр. – 2018. –

Режим доступа до ресурсу: <https://habr.com/ru/post/358804/>.

23. Advanced iOS App Architecture (First Edition): Real-world app architecture in Swift / Rene Cacheaux & Josh Berlin, 2018. – Raywenderlich.
24. Data Structures and Algorithms in Swift, 2017. – Raywenderlich.
25. Design Patterns by Tutorials, 2018. – Raywenderlich.
26. Concurrency by Tutorials, 2017. – Raywenderlich.
27. iOS 10 Programming Fundamentals with Swift: Swift, Xcode, and Cocoa Basics / Matt Neuburg, 2017.
28. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch, 2018.
29. Основы разработки приложений под iOS и macOS / Василий Усов, 2018.
30. Разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK / Молли Маскри, 2017.
31. Swift Programming Language / Apple, 2019
32. Human Interface Guidelines / Apple, 2019.
33. 100 Things Every Designer Needs to Know About People / Susan Weinschenk, 2021