МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ГУЦУЛЯК ДАР'Я ВАЛЕРІЇВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р техн. наук, професор
_____Наталія ВЕСЕЛОВСЬКА
«____»_____ 2026 р.

ПРОГРАМНА РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ СИСТЕМИ УПРАВЛІННЯ З КЛІЄНТАМИ ОНЛАЙН-МАГАЗИНУ МЕБЛІВ

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:
Ірина ФРИЗ, старший викладач кафедри
інформаційних технологій
канд. фіз.-мат. наук
_____

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2026

АНОТАЦІЯ

Гуцуляк Д. В. Програмна реалізація взаємодії системи управління з клієнтами онлайн – магазину меблів. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

Кваліфікаційну роботу присвячено проєктуванню та реалізації програмного модуля CRM-системи для інтернет-магазину «Мебельки». Проаналізовано архітектурні аналоги, сформовано вимоги до ПЗ та спроєктовано реляційну базу даних у СУБД PostgreSQL. Розроблено архітектуру модуля на основі веб-технологій, Node.js та React.js. Розглянуто інтеграційні API для збору поведінкових маркерів та кейс міграції користувачів між маркетинговими категоріями. Проаналізовано детермінований RFM-аналіз та кластеризацію K-середніх на основі Евклідової відстані. Проведено функціональне тестування, що підтвердило надійність тригерів реального часу та працездатність системи.

Ключові слова: CRM-система, електронна комерція, PostgreSQL, FastAPI, React.js, RFM-аналіз, алгоритм K-середніх, кластеризація, тригери реального часу.

63 ст., 16 рис., 5 табл., 2 дод. 41 джерело.

ABSTRACT

Hutsuliak D. V. Software Implementation of the Interaction of the Management System with Customers of an Online Furniture Store. Specialty 122 «Computer Science». Educational program «Computer Science». Vasyl' Stus Donetsk National University, Vinnytsia, 2026.

The qualification thesis is devoted to the design and implementation of a CRM system software module for the Mebelki online store. Architectural analogues are analyzed, software requirements are formulated, and a relational database is designed in the PostgreSQL DBMS. The module architecture based on web technologies, Node.js, and React.js is developed. Integration APIs for collecting behavioral markers and a use case of user migration between marketing categories are considered. Deterministic RFM analysis and K-means clustering based on Euclidean distance are analyzed. Functional testing was conducted, confirming the reliability of real-time triggers and the operational capacity of the system.

Keywords: CRM system, e-commerce, PostgreSQL, FastAPI, React.js, RFM analysis, K-means algorithm, clustering, real-time triggers.

63 p., 16 figs., 5 tabs., 2 app. 41 sources.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CRM (Customer Relationship Management) – система управління взаємовідносинами з клієнтами;

RFM (Recency, Frequency, Monetary) – метод аналітичної сегментації клієнтів за давністю (R), частотою (F) та сумою (M) їхніх покупок;

K-середніх (K-means) – алгоритм машинного навчання для автоматичного поділу клієнтської бази на групи (кластери) за рівнем схожості;

LTV (Lifetime Value) – сукупний чистий прибуток, який бізнес отримує від одного покупця за весь час співпраці;

CAC (Customer Acquisition Cost) – вартість залучення одного клієнта; сума всіх маркетингових та операційних витрат на одного нового покупця;

ERP (Enterprise Resource Planning) – автоматизована система планування ресурсів підприємства;

API (Application Programming Interface) – інтерфейс прикладного програмування;

Elasticsearch – аналітична пошукова система для організації швидкого повнотекстового пошуку у великих масивах даних;

Redis – база даних в оперативній пам’яті, що використовується для кешування та миттєвого доступу до даних;

FastAPI – швидкий асинхронний веб-фреймворк на Python для розробки високонавантажених та продуктивних API;

Оркестрація процесів – централізоване автоматизоване керування та координація послідовності виконання завдань або мікросервісів;

Маршрутизація запитів – процес визначення й спрямування вхідних HTTP-запитів до відповідних функцій-обробників на сервері;

ASGI (Asynchronous Server Gateway Interface) – стандарт асинхронного шлюзу для одночасної обробки веб-сервером великої кількості запитів на Python;

OpenAPI – єдиний міжнародний стандарт (специфікація) для проєктування та опису структури RESTful API;

Node.js – серверне середовище виконання коду на мові JavaScript для побудови масштабованих вебзастосунків;

JSON (JavaScript Object Notation) – стандарт для структурованого обміну даними між сервером та клієнтом;

NPM (Node Package Manager) – реєстр бібліотек для платформи Node.js.;

REST (Representational State Transfer) – архітектурний стиль взаємодії веб-сервісів за допомогою стандартних методів протоколу HTTP;

Low-code/No-code – методи створення ПЗ за допомогою візуальних конструкторів із мінімальним написанням коду (Low-code) або без нього (No-code);

PostgreSQL – об’єктно-реляційна СУБД із підтримкою SQL (мови управління даними та запитів в базах даних, транзакцій;

MySQL –реляційна СУБД з відкритим кодом, оптимізована для швидкої роботи у стандартних вебзастосунках;

MongoDB – нереляційна (NoSQL) документ-орієнтована СУБД, яка зберігає дані у гнучкому JSON-подібному форматі;

React.js – JavaScript-бібліотека від Meta для розробки динамічних інтерфейсів користувача (Frontend) на основі компонентів.

ВСТУП

Актуальність теми. У сучасних умовах цифровізації економіки інтернет-магазини стали невід'ємною частиною комерційної діяльності, забезпечуючи зручність пошуку та придбання товарів для широкого кола користувачів. Проте висока конкуренція у сфері електронної комерції вимагає від бізнесу не лише наявності онлайн-майданчика, а й глибокого розуміння потреб споживачів [1]. Системи управління взаємовідносинами з клієнтами (CRM) дозволяють автоматизувати взаємодію з покупцями, підвищити їхню лояльність та оптимізувати маркетингові стратегії на основі аналізу великих масивів даних. Розробка спеціалізованого програмного забезпечення, окремого програмного модуля CRM, адаптованого під специфіку онлайн-торгівлі, є актуальним завданням для забезпечення конкурентоспроможності сучасних підприємств [2-3].

Мета дослідження – розробка програмного забезпечення для покращення ефективності управління взаємовідносинами клієнтами у сегменті електронної комерції.

Завдання дослідження:

- проаналізувати особливості систем управління взаємовідносинами з клієнтами, кейси їх використання;
- виконати аналіз сучасних технологій веброзробки та інструментів для сегментації даних;
- спроєктувати структуру програмного модуля для меблевого магазину, бази даних для надійного збереження транзакційної та поведінкової інформації щодо взаємовідносин з клієнтами;
- реалізувати модуль збору даних, сегментації клієнтів та візуалізації статусу клієнтів;
- провести тестування розробленого програмного забезпечення.

Об'єкт дослідження – процеси автоматизації процедур взаємодії з клієнтами в галузі електронної комерції на прикладі магазину «Мебельки».

Предмет дослідження – методи, архітектурні рішення та алгоритми сегментації у складі програмного модуля управління взаємовідносинами з клієнтами.

Практичне значення результатів. Розроблене програмне забезпечення може бути використане власниками онлайн-магазинів меблів та адаптовано для автоматизації маркетингових розсилок, прогнозування відтоку клієнтів та формування персоналізованих пропозицій в галузі електронної комерції, що безпосередньо впливає на зростання прибутку та ефективність бізнесу.

Апробація отриманих результатів. Результати дослідження доповідалися на VII Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих вчених «Прикладні інформаційні технології», 15 травня 2026 р., Донецький національний університет імені Василя Стуса, м. Вінниця, тема доповіді: «Програмний модуль управління взаємовідносинами з клієнтами онлайн магазину електронної комерції».

Структура роботи. Робота складається зі вступу, трьох розділів, висновків та списку використаних джерел. У першому розділі проведено теоретичний аналіз систем управління клієнтами та аналіз інформаційних технологій. У другому розділі представлено проєктування архітектури системи, структури програмного забезпечення та логіки алгоритмів сегментації. У третьому розділі висвітлено практичну реалізацію програмного продукту та результати його тестування. Робота містить 63 сторінки тексту, 16 рисунків, 5 таблиць, 2 додатки, 40 посилань на літературу та інформаційні джерела, 2 додатки.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНОЛОГІЙ РОЗРОБКИ CRM-СИСТЕМИ

1.1 Основні поняття та роль CRM-систем у сегменті електронної комерції

У сучасних умовах розвитку інтернет-торгівлі стратегія управління взаємовідносинами з клієнтами (Customer Relationship Management, CRM) стає визначальним фактором стабільності бізнесу. CRM – це не лише програмний продукт, а комплексна бізнес-стратегія, спрямована на побудову довгострокових та взаємовигідних відносин із клієнтами за допомогою збору, зберігання та аналізу інформації про їхню поведінку [2].

Для онлайн-магазинів використання CRM-систем дозволяє вирішити наступні завдання:

- Централізація даних – збір контактної інформації, історії замовлень та звернень до служби підтримки в єдиному профілі клієнта.
- Персоналізація маркетингу – формування індивідуальних пропозицій на основі попередніх покупок.
- Автоматизація воронки продажів – контроль шляху клієнта від першого відвідування сайту до завершення транзакції та повторних продажів [3].

Класична архітектура CRM для e-commerce передбачає наявність трьох ключових компонентів: операційного (автоматизація маркетингу та сервісу), аналітичного (обробка даних для виявлення закономірностей) та колаборaційного (взаємодія з клієнтом через різні канали зв’язку: e-mail, чат-боти, телефонія).

Концепція управління взаємовідносинами з клієнтами в межах електронної комерції розглядається як інтегрована інформаційна система, що забезпечує автоматизацію стратегій взаємодії з покупцями на всіх етапах життєвого циклу:

від залучення до післяпродажного обслуговування. На відміну від стандартних систем обліку, CRM для електронної комерції орієнтована на максимізацію показника LTV (Lifetime Value) та зниження вартості залучення клієнта (CAC) [2].

На рисунку 1.1 представлено загальну структуру системи управління взаємовідносинами з клієнтами.

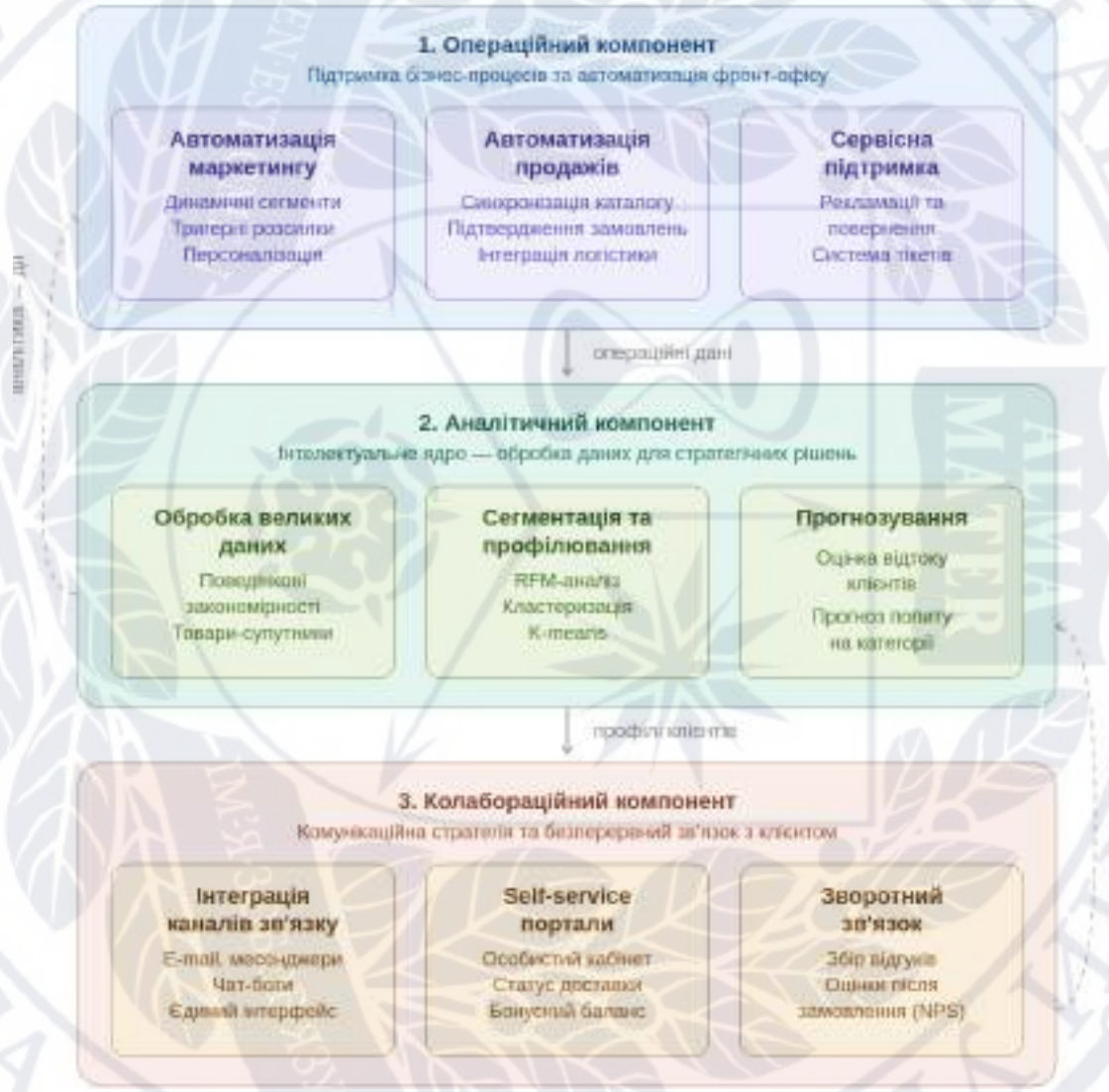

Рисунок 1.1 – Загальна структура системи управління взаємовідносинами з клієнтами

Розглянемо основні компоненти системи управління взаємовідносинами з клієнтами (на основі аналізу літературних джерел та популярних CRM [3;5-9].

1. Операційний компонент. Даний рівень забезпечує безпосередню підтримку бізнес-процесів та автоматизацію рутинних операцій фронт-офісу. В контексті інтернет-магазину він охоплює:

- автоматизація маркетингу – управління динамічними сегментами для розсилок, тригерні повідомлення при залишенні кошика, персоналізація контенту на сайті;
- автоматизація продажів – синхронізація з каталогом товарів, автоматичне підтвердження замовлень, управління статусами оплат та інтеграція з логістичними сервісами;
- сервісна підтримка – обробка рекламацій, повернень та технічна підтримка користувачів.

2. Аналітичний компонент. Це «інтелектуальне ядро» системи, яке не має прямого контакту з клієнтом, але обробляє накопичені операційні дані для прийняття стратегічних рішень. Його завдання включають такі процедури:

- обробка великих даних – виявлення прихованих закономірностей у поведінці користувачів (наприклад, визначення товарів-супутників для комплексних покупок);
- сегментація та профілювання – використання математичних моделей (RFM-аналіз, кластеризація K-середніх) для групування клієнтів за рівнем доходу, частотою покупок та лояльністю;
- прогнозування – оцінка ймовірності відтоку клієнтів та прогнозування майбутнього попиту на певні категорії товарів.

3. Колобораційний компонент. Даний рівень відповідає за комунікаційну стратегію та забезпечення безперервного зв’язку з клієнтом через комунікаційну модель каналів зв’язку. Він реалізується через інтеграцію каналів зв'язку – об’єднання в єдиному інтерфейсі діалогів з e-mail, месенджерів, чат-ботів. Через особисті кабінети клієнтів можна самостійно відстежувати історію покупок,

бонусний баланс та статус доставки, що знижує навантаження на операторів також є реалізацією компонентів. В системі реалізовано зворотний зв'язок – автоматизований збір відгуків та оцінок після отримання замовлення для розрахунку індексу споживчої лояльності дозволяє збирати дані для сегментації клієнтів [5-6].

Програмний модуль управління взаємовідносинами з клієнтами можна представити як багаторівневу програмну систему, що забезпечує автоматизацію процесів взаємодії організації з її клієнтською базою. Архітектурно модуль будується за принципом розподілу відповідальності між чітко визначеними шарами, кожен з яких виконує відособлену функцію та взаємодіє із суміжними шарами через стандартизовані інтерфейси. На рисунку 1.2 представлено таку багаторівневу програмну систему. Вона відповідає кращим практикам у напряму взаємовідносин з клієнтами в світі та Україні [5;6]

Верхній рівень системи – рівень представлення – забезпечує взаємодію з кінцевими користувачами через веб-інтерфейс, мобільні додатки та програмні інтерфейси для зовнішніх клієнтів. Саме на цьому рівні формуються запити, які передаються для опрацювання до шару бізнес-логіки.

Шар бізнес-логіки є центральним компонентом модуля. Він відповідає за виконання бізнес-правил, оркестрацію процесів та маршрутизацію запитів між функціональними підсистемами. На цьому рівні реалізуються умови переходів між станами сутностей та правила автоматизованого реагування на події.

Функціональна частина модуля охоплює шість взаємопов'язаних підсистем. Підсистема управління контактами забезпечує зберігання та опрацювання даних про клієнтів і потенційних покупців, включно з їх сегментацією. Підсистема продажів реалізує управління воронкою угод та прогнозування доходів. Підсистема маркетингу забезпечує планування і проведення комунікаційних кампаній, у тому числі засобами автоматизації розсилок. Підсистема підтримки клієнтів обробляє звернення у форматі тікетів, контролює дотримання угод про

рівень обслуговування та веде базу знань. Підсистема аналітики формує зведені звіти, дашборди та ключові показники ефективності. Підсистема активностей фіксує взаємодії з клієнтами — дзвінки, зустрічі, завдання та нотатки.

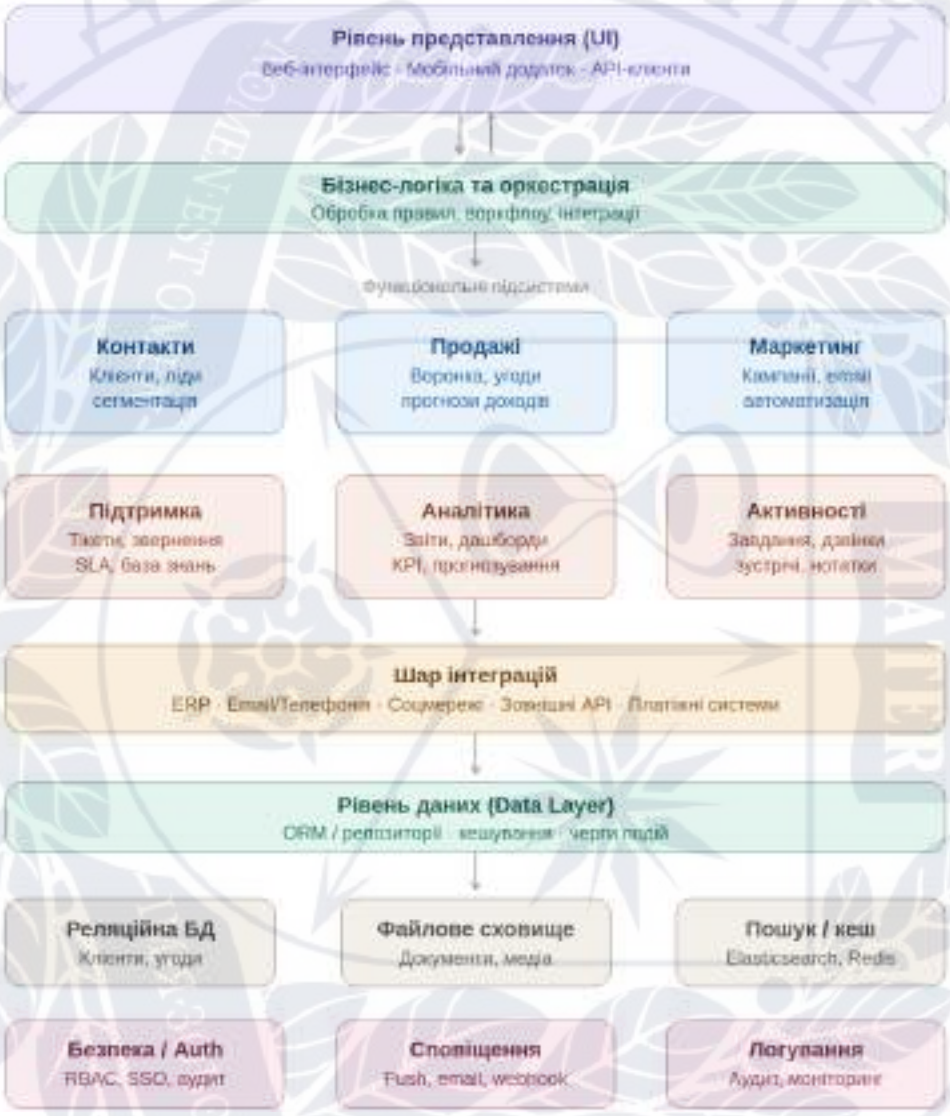

Рисунок 1.2 – Багаторівнева система управління взаємовідносинами з клієнтами

Шар інтеграцій забезпечує взаємодію модуля із зовнішніми системами: корпоративними ERP-рішеннями, поштовими та телефонними сервісами, соціальними мережами і платіжними системами. Комунікація здійснюється через стандартизовані API.

Рівень даних абстрагує роботу зі сховищами інформації. Транзакційні дані зберігаються в реляційній базі даних, документи та медіафайли — у виділеному файловому сховищі, тоді як для забезпечення швидкого повнотекстового пошуку і кешування доцільно застосовувати спеціалізовані рішення, такі як Elasticsearch та Redis.

Наскрізними для всієї архітектури є три підсистеми, зокрема безпеки, сповіщень, логування. Підсистема безпеки реалізує рольову модель доступу, підтримку єдиного входу та ведення журналу аудиту. Підсистема сповіщень забезпечує доставку повідомлень через різні канали. Підсистема логування здійснює моніторинг стану системи та фіксацію подій для цілей аудиту та діагностики.

Таким чином, ефективна побудова програмного забезпечення CRM-системи для онлайн-магазину вимагає гармонійної інтеграції всіх трьох компонентів. Операційний рівень забезпечує збір первинних даних, аналітичний перетворює їх на корисні інсайти, а колобораційний дозволяє реалізувати ці інсайти у формі персоналізованої комунікації, що безпосередньо впливає на конверсію та прибутковість підприємства.

1.2 Аналіз сучасних технологій для створення CRM систем

Проєктування сучасного програмного забезпечення для управління клієнтською базою в електронній комерції базується на принципах розподілу відповідальності між клієнтською та серверною частинами. Основним архітектурним шаблоном для реалізації CRM-системи обрано клієнт-серверну модель з використанням прикладного програмного інтерфейсу [9]. Такий підхід забезпечує незалежність розробки фронтенд- та бекенд-складових, що дозволяє оптимізувати навантаження на систему та забезпечити високу гнучкість для подальшої модернізації окремих модулів.

Для побудови серверної частини системи можна використати різноманітні фреймворки, наприклад FastAPI на мові програмування Python або інших фреймворків. Даний вибір зумовлений потребою в забезпеченні високої пропускної здатності при обробці запитів від адміністративної панелі та зовнішніх систем. На відміну від традиційних синхронних фреймворків, FastAPI базується на стандарті ASGI, що дозволяє обробляти вхідні з'єднання асинхронно. Це суттєво знижує час очікування відповіді сервера при виконанні операцій з базою даних або зверненнях до сторонніх сервісів. Важливою перевагою обраного фреймворку є автоматична генерація документації за стандартом OpenAPI, що спрощує процес налагодження взаємодії між фронтендом та бекендом, забезпечуючи чітку структуру передачі об'єктів даних [10; 11].

У процесі аналізу інструментарію для побудови серверної частини сучасних інформаційних систем особливе місце посідає Node.js. Дана технологія не є окремою мовою програмування, а представляє собою програмне середовище виконання мови JavaScript на стороні сервера, побудоване на рушії V8. Основною архітектурною особливістю Node.js є використання подійно-орієнтованої моделі та неблокуючого введення-виведення, що дозволяє обробляти велику кількість одночасних з'єднань з мінімальними витратами системних ресурсів [12;13].

Для CRM-систем у сфері електронної комерції використання Node.js забезпечує високу ефективність під час реалізації функціоналу, що працює в режимі реального часу. Це стосується насамперед модулів миттєвих повідомлень, систем сповіщень про зміну статусів замовлень та інтерактивних панелей моніторингу активності користувачів. Завдяки використанню єдиної мови програмування як на фронтенді, так і на бекенді, значно спрощується підтримка коду та забезпечується висока швидкість передачі об'єктів даних у форматі JSON.

Проте під час порівнянні з іншими технологіями слід враховувати специфіку обчислювальних процесів. Node.js демонструє найкращі результати в операціях, пов'язаних з інтенсивним мережевим обміном та запитами до баз даних, але може

поступатися у продуктивності при виконанні складних математичних розрахунків, які потребують значних ресурсів центрального процесора. В контексті аналітичних модулів CRM, де передбачається виконання алгоритмів сегментації та кластеризації великих масивів даних, Node.js часто потребує винесення таких завдань в окремі мікросервіси або використання додаткових бібліотек.

Незважаючи на зазначені обмеження, величезна бібліотека пакетів NPM надає розробнику доступ до численних готових рішень для інтеграції з платіжними шлюзами, поштовими сервісами та інструментами аналітики. Таким чином, середовище Node.js розглядається як потужна альтернатива для побудови масштабованих і швидких серверних компонентів, що забезпечують оперативну взаємодію з клієнтами в динамічному середовищі інтернет-торгівлі.

Архітектура інтерфейсу користувача будується на основі бібліотеки React.js, яка реалізує декларативний підхід до створення компонентів. Використання даної технології дозволяє побудувати систему за принципом створення веб-системи та перенесенні логіки відображення на сторону клієнта, що зменшує трафік між браузером та сервером. Завдяки використанню віртуального об’єктного дерева документів, React.js забезпечує високу швидкість рендерингу складних інтерфейсів, що є критичним для CRM-систем з великою кількістю динамічних таблиць, графіків та форм редагування даних. Модульна структура компонентів дозволяє створити масштабований інтерфейс, який легко адаптується під різні типи пристроїв [14].

Взаємодія між фронтендом та бекендом реалізується через архітектурний стиль REST. Це передбачає використання стандартних протоколів передачі даних, де кожен запит містить усю необхідну інформацію для його обробки, що дозволяє серверу залишатися безстатусною системою [15].

Для збереження даних доцільно використовувати реляційні моделі на базі СУБД [16].

PostgreSQL, яка забезпечує цілісність інформації та підтримує складні зв’язки між об’єктами, що є характерним для систем управління взаємовідносинами. Поєднання асинхронного бекенд-фреймворку з компонентною фронтенд-бібліотекою створює стійку технологічну базу для розробки високоефективного програмного забезпечення, здатного стабільно функціонувати в умовах змінного навантаження, властивого сфері інтернет-торгівлі [17].

Вибір системи управління базами даних є фундаментальним етапом проектування CRM-системи, оскільки саме цей компонент відповідає за надійність збереження персональних даних, історію транзакцій та швидкість формування аналітичних вибірок. У сучасній практиці розробки інформаційних систем для електронної комерції розглядаються два основні підходи до організації даних: реляційний та нереляційний [16].

Реляційні СУБД, такі як PostgreSQL та MySQL, базуються на структурі зв’язаних таблиць та забезпечують цілісність даних через механізми зовнішніх ключів [16 – 1]. PostgreSQL вважається найбільш прогресивним рішенням серед безкоштовних систем завдяки повній відповідності стандартам транзакційності та можливості обробки складних аналітичних запитів. Вона дозволяє ефективно працювати з великими обсягами інформації, підтримуючи як традиційні табличні дані, так і об’єкти в форматі JSON. Це особливо важливо для CRM-систем, де профіль клієнта може містити варіативні атрибути, що не вкладаються в жорстку схему.

Нереляційні системи, яскравим представником яких є MongoDB, пропонують документ-орієнтовану модель збереження даних [20]. Основною перевагою таких систем є висока швидкість горизонтального масштабування та відсутність жорсткої схеми даних.

В таблиці 1.1 представлені порівняльні характеристики СУБД, сформовані з використанням відкритих джерел за типом системи, структурою даних, їх цілісністю, масштабування, роботи з аналітичними запитами тощо.

Таблиця 1.1 Порівняння СУБД

Критерій порівняння	**PostgreSQL**	**MySQL**	**MongoDB**
Тип системи (архітектура)	Реляційна (об’єктно-реляційна)	Реляційна	Нереляційна (документ-орієнтована)
Структура даних	Таблиці зі зв’язками, підтримка JSON	Таблиці зі зв'язками (зовнішні ключі)	Документи (BSON/JSON), без жорсткої схеми
Цілісність даних та транзакційність	Повна відповідність ACID, висока суворість	Підтримка ACID (з рушієм InnoDB)	Базова (обмежена на рівні документів)
Масштабування	Переважно вертикальне	Переважно вертикальне	Високе горизонтальне (шардинг з коробки)
Швидкість роботи	Висока при складних запитах та операціях запису	Висока при простих операціях читання	Дуже висока на простих операціях запису/читання
Робота зі складними аналітичними запитами	Відмінна (підтримка складних об’єднань JOIN та віконних функцій)	Середня (можливе просідання продуктивності при важких JOIN)	Обмежена (потребує побудови конвеєрів агрегації)
Основна роль в архітектурі CRM / онлайн-магазину	Основна БД: збереження транзакцій, профілів користувачів, генерація звітів	Альтернативна основна БД для стандартних інтернет-магазинів	Зберігання каталогів товарів з варіативними атрибутами

У порівнянні з MySQL, PostgreSQL демонструє кращу стабільність при виконанні важких операцій запису та складних об’єднань декількох таблиць, що є

типовим процесом при генерації звітів про лояльність. Це дозволяє швидко вносити зміни в структуру об’єктів без необхідності зупинки системи. Проте для CRM-систем, де важливою є точність фінансових розрахунків та наявність чітких зв’язків між замовленнями, товарами та клієнтами, використання NoSQL рішень як основної бази даних може призвести до надмірної денормалізації та складнощів при контролі цілісності інформації.

Окрему категорію становлять системи збереження даних у пам’яті, такі як Redis [21]. Вони характеризуються надзвичайно високою швидкістю доступу до інформації за рахунок відмови від постійного звернення до дискової підсистеми. У архітектурі CRM-системи Redis найчастіше використовується як допоміжний інструмент для кешування результатів аналітичних розрахунків або зберігання сесій користувачів, що дозволяє суттєво розвантажити основну реляційну базу даних.

Таким чином, для реалізації проектованого програмного забезпечення найбільш доцільним є використання PostgreSQL як основної системи управління базами даних. Такий вибір забезпечує необхідний баланс між суворістю збереження транзакційної інформації та гнучкістю при роботі з аналітичними даними, що є ключовим фактором для стабільного функціонування онлайн-магазину в довгостроковій перспективі.

1.3 Порівняльний аналіз аналогів програмного модуля управління взаємовідносинами з клієнтами

На сьогоднішній день ринок CRM-систем пропонує як універсальні SaaS-платформи, так і спеціалізовані модулі для конкретних платформ [22; 23].

На сучасному ринку програмного забезпечення представлено широку номенклатуру систем управління взаємовідносинами з клієнтами, кожна з яких має специфічний набір інструментів для автоматизації комерційної діяльності. Для

обґрунтування доцільності розробки власного програмного продукту необхідно провести критичний аналіз найбільш розповсюджених аналогів, що застосовуються в сегменті електронної торгівлі.

Одним із лідерів галузі є система Salesforce (США), яка функціонує за моделлю програмного забезпечення як послуги [24]. Дана платформа пропонує глибоку інтеграцію з різноманітними торговельними майданчиками та містить потужні інструменти для прогнозування продажів.

Salesforce представляє собою хмарну екосистему, яка використовує модель спільних ресурсів, що дозволяє багатьом організаціям використовувати спільні обчислювальні ресурси та інфраструктуру при повному логічному розділенні даних. Це забезпечує високу масштабованість та автоматичне оновлення програмного забезпечення без участі кінцевого користувача [24].

Функціональна структура системи базується на концепції Customer 360, що передбачає створення єдиного профілю клієнта шляхом інтеграції даних з різних каналів: маркетингу, продажів, сервісного обслуговування та аналітики. В контексті електронної комерції ключовим модулем, який дозволяє автоматизувати процес покупки та забезпечує персоналізацію контенту на основі штучного інтелекту Einstein. Даний алгоритм аналізує історію переглядів та транзакцій у реальному часі, що дозволяє системі автоматично генерувати рекомендації та прогнозувати ймовірність повторних замовлень.

Важливою перевагою Salesforce є наявність власної мови програмування Apex, яка за синтаксисом подібна до Java. Це надає розробникам можливість створювати унікальну бізнес-логіку та специфічні інтерфейси, що виходять за межі стандартних налаштувань. Крім того, екосистема AppExchange пропонує тисячі готових рішень для інтеграції з логістичними операторами, платіжними шлюзами та зовнішніми базами даних, що робить платформу надзвичайно гнучким інструментом для великих торговельних мереж.

Незважаючи на лідерські позиції, Salesforce має ряд особливостей, що обмежують її використання для малого бізнесу та стартапів. По-перше, вартість ліцензій за кожного користувача є однією з найвищих на ринку, що створює значне фінансове навантаження при розширенні штату менеджерів. По-друге, система характеризується високим порогом входження: для ефективного впровадження та підтримки необхідні кваліфіковані адміністратори та розробники. По-третє, орієнтованість на універсальність іноді призводить до надмірної складності інтерфейсу для вирішення простих завдань онлайн-торгівлі. Таким чином, Salesforce виступає потужним еталоном для аналізу при розробці спеціалізованих CRM-систем, проте залишає відкритою нішу для більш доступних та спеціалізованих програмних продуктів.

На рисунку 1.3 представлено вигляд екрану роботи системи Salesforce.

Рисунок 1.3 – Вигляд екрану роботи системи Salesforce

Головною перевагою Salesforce є високий рівень кастомізації та можливість масштабування під потреби великих корпорацій. Проте для малого та середнього бізнесу використання цієї системи пов’язане з надмірними фінансовими

витратами на придбання ліцензій та складністю процесу первинного налаштування, що потребує залучення сертифікованих спеціалістів. Крім того, надмірний функціонал часто перевантажує інтерфейс користувача, уповільнюючи роботу операторів.

Вагоме місце на міжнародному ринку посідає українська платформа Creatio [25]. Вона базується на low-code технологіях, що дозволяє адаптувати бізнес-процеси без глибокого програмування. Система вирізняється потужними інструментами для управління маркетингом та продажами в єдиному середовищі.

Creatio є провідною українською розробкою в класі систем управління взаємовідносинами з клієнтами, яка отримала міжнародне визнання завдяки впровадженню концепції low-code/no-code. Основна ідеологія платформи полягає в наданні користувачам можливості конструювати бізнес-процеси та інтерфейси без глибокого знання мов програмування, що значно скорочує термін впровадження системи (Рисунок 1.4).

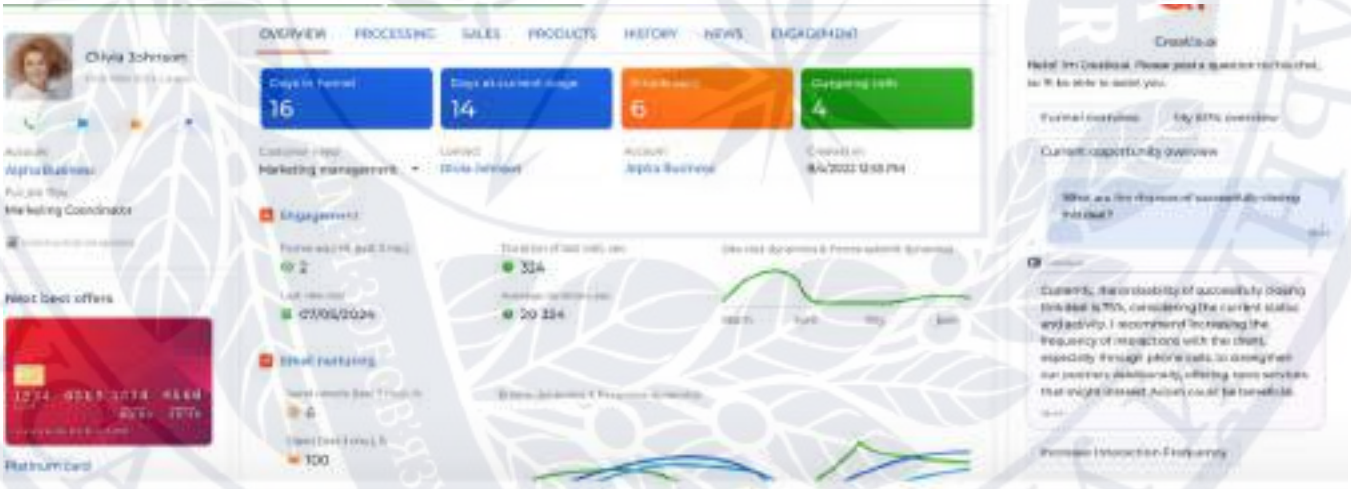

Рисунок 1.4 – Вигляд екрану роботи системи Creatio

Архітектурно Creatio побудована на базі стека .NET, що забезпечує високу продуктивність та сумісність з широким спектром корпоративних технологій.

Система базується на трикомпонентній структурі, яка об’єднує модулі для автоматизації маркетингу, продажів та сервісного обслуговування на єдиній

платформі. Для сфери електронної комерції особливе значення має інтегрований двигун керування бізнес-процесами (BPM), який працює за стандартом BPMN 2.0. Це дозволяє детально моделювати шлях клієнта від моменту першого відвідування веб-сайту до завершення транзакції та післяпродажного супроводу, автоматизуючи переходи між етапами воронки продажів.

Важливою особливістю платформи є її аналітичний модуль, який використовує інструменти машинного навчання для предиктивного аналізу. Creatio дозволяє будувати моделі для прогнозування ймовірності успішного закриття угод, оцінки лідів та автоматичного формування рекомендацій щодо наступних кроків взаємодії з клієнтом. На відміну від багатьох аналогів, система пропонує гнучкі інструменти візуалізації даних, що дозволяють створювати складні аналітичні панелі для моніторингу ключових показників ефективності онлайн-магазину в реальному часі.

З точки зору розробника, Creatio підтримує складні інтеграційні сценарії через використання REST API, що забезпечує безшовну синхронізацію з інтернет-магазинами, платіжними системами та службами доставки. Платформа підтримує роботу як у хмарному середовищі, так і на власних серверах замовника, що є важливим для компаній з суворими вимогами до безпеки та локалізації даних.

Проте, незважаючи на технологічну досконалість, Creatio орієнтована насамперед на середній та великий бізнес. Це зумовлює складну структуру ліцензування та потребу в системному підході до навчання персоналу. Для малих підприємств електронної комерції вартість впровадження та підтримки такої платформи може бути надмірною, що підтверджує актуальність розробки більш легких, спеціалізованих CRM-рішень, адаптованих під потреби конкретних торгових майданчиків. Таким чином, Creatio виступає як приклад високоефективної системи з потужним аналітичним потенціалом, на досвід якої доцільно спиратися при проектуванні архітектури власного програмного

Основним недоліком Creatio для невеликих проектів електронної комерції є орієнтованість на великий корпоративний сегмент (Enterprise), що відображається на вартості володіння та складності архітектури системи для вирішення локальних завдань сегментації.

Європейським аналогом, що широко використовується малим та середнім бізнесом, є естонська система Pipedrive [26]. Зокрема, реалізація специфічних алгоритмів кластеризації або складних моделей сегментації потребує підключення сторонніх сервісів, що збільшує підсумкову вартість програмного рішення. Вигляд екрану роботи системи представлено на рисунку 1.5.

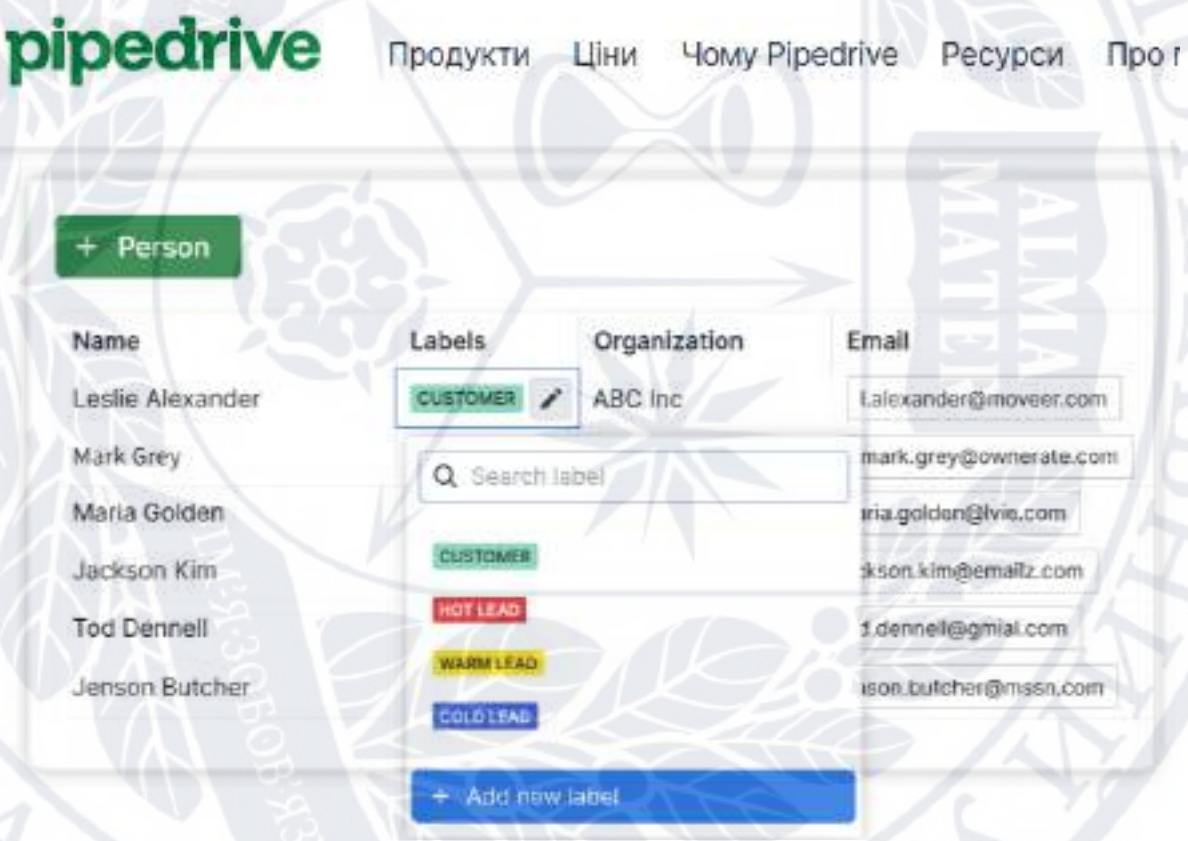

Рисунок 1.5 – Вигляд екрану роботи системи Pipedrive

Вона характеризується високим рівнем зручності інтерфейсу та фокусом на візуалізації воронки продажів. Pipedrive забезпечує ефективне управління угодами та легку інтеграцію з поштовими сервісами. Проте аналітичні можливості системи щодо глибокого статистичного аналізу клієнтської бази є обмеженими. Pipedrive

представляє собою хмарне програмне рішення естонського походження, розроблене для автоматизації процесів управління продажами та взаємодії з клієнтами. Архітектура системи базується на принципах максимальної візуалізації робочих процесів та мінімізації часу на введення адміністративної інформації. Основною концептуальною відмінністю даного продукту є фокус на управлінні діями, що безпосередньо ведуть до завершення транзакції, а не на пасивному обліку контактних даних.

Інтерфейс системи побудований навколо візуальної моделі конвеєра продажів, що дозволяє менеджерам оперативно відстежувати рух кожного потенційного замовлення між етапами воронки. Така структура забезпечує прозорість бізнес-процесів та дозволяє виявляти вузькі місця в циклі обслуговування клієнтів інтернет-магазину. Pipedrive пропонує інструментарій для автоматизації повторюваних завдань, включаючи планування контактів, розсилку персоналізованих електронних листів та генерацію типових документів, що суттєво знижує навантаження на операційний персонал.

Важливою перевагою системи є її інтеграційна здатність. Завдяки відкритому програмному інтерфейсу та наявності широкого каталогу готових додатків у маркетплейсі, Pipedrive легко синхронізується з популярними платформами електронної комерції, системами аналітики та інструментами автоматизації маркетингу. Це дозволяє використовувати систему як центральний вузол збору інформації про клієнтську активність без необхідності розробки складних проміжних рішень.

Проте аналіз експлуатаційних можливостей Pipedrive виявляє певну обмеженість у частині поглибленої аналітики даних.

Система орієнтована на оперативне управління поточною діяльністю, тому базовий функціонал не містить вбудованих інструментів для проведення складного статистичного аналізу клієнтської бази або сегментації на основі методів машинного навчання. Реалізація предиктивних моделей або кластерного

аналізу в межах даної платформи потребує експорту даних у зовнішні аналітичні середовища або придбання додаткових високовартісних модулів.

Для сегмента електронної торгівлі, де важливим є розуміння прихованих закономірностей у поведінці споживачів, такий підхід може бути недостатнім. Також варто зазначити, що модель ціноутворення Pipedrive передбачає оплату за кожного активного користувача, що при великому штаті менеджерів з продажу може бути економічно недоцільним для підприємств з низькою маржинальністю товарів. Таким чином, Pipedrive є ефективним прикладом ергономічного та функціонального дизайну, проте потреба в інтегрованих інтелектуальних модулях аналізу залишається актуальною передумовою для розробки спеціалізованого програмного забезпечення.

Порівняльна характеристика наведених систем представлена у таблиці 1.2. Проведений порівняльний аналіз дозволяє зробити висновок, що незважаючи на наявність потужних міжнародних та вітчизняних рішень, існує потреба у розробці спеціалізованого програмного забезпечення, яке б поєднувало в собі лаконічність інтерфейсу, доступність та наявність інтегрованих модулів автоматизованої сегментації на основі поведінкових даних.

Головний недолік переважної кількості систем управління взаємовідносинами з клієнтами – їх орієнтація на повний стек функцій управління клієнтами, який є надлишковим для нішового сегменту, а також для систем малого бізнесу [24-29]. Крім того, адаптація до особливостей бізнесу та поступова розробка необхідних функцій відсутня в закритих системах або потребує додаткових коштів для придбання нових модулів та супроводження процесів впровадження системи. Доцільним також є інтеграція аналітичних модулів взаємовідносин з клієнтами в соціальних мережах. Виконаний порівняльний аналіз свідчить про різноманіття систем управління взаємовідносинами з клієнтами, а також свідчить про достатньо високу ціну для повнофункціональних систем та задачі інтеграції їх з різноманітними модулями аналітики.

Таблиця 1.2 Порівняльний аналіз CRM-систем для онлайн-торгівлі

Критерії порівняння	**Salesforce**	**Creatio**	**Pipedrive**
Країна походження	США	Україна	Естонія
Основна цільова аудиторія	Великі корпорації (Enterprise)	Середній та великий бізнес	Малий та середній бізнес
Модель розгортання	Хмарна (SaaS)	Хмарна та локальна (On-site)	Хмарна (SaaS)
Ключова технологічна особливість	Мультитенантна архітектура, мова Apex	Low-code/No-code платформа, BPM	Візуалізація воронки продажів (Pipelines)
Аналітичні можливості	Високі (AI Einstein)	Високі (ML-моделі прогнозування)	Базові (операційна звітність)
Складність впровадження	Висока (потребує сертифікованих спеціалістів)	Середня/Висока (через складність процесів)	Низька (інтуїтивно зрозумілий інтерфейс)
Інтеграція з e-commerce	Глибока (Commerce Cloud, AppExchange)	Широка (через REST API та готові конектори)	Середня (через Marketplace та API)
Вартість володіння	Висока	Середня/Висока	Помірна

Проєктована CRM-система покликана заповнити цю нішу, пропонуючи ефективний інструментарій для аналізу клієнтської бази без необхідності впровадження громіздких корпоративних платформ.

Функціональні вимоги до модуля визначають обов’язкові сервіси та сценарії поведінки системи. Програмне забезпечення повинно автоматично фіксувати й зберігати історію успішних транзакцій, а також логувати поточні дії покупців на сайті. Аналітичний блок має забезпечувати автоматичний розрахунок маркетингових метрик для кожного профілю та групування споживачів на задану кількість кластерів для виявлення прихованих закономірностей. Такі дані та

обробка може виконуватись спеціальними аналітичними модулями (наприклад, інструменти аналізу Google). У межах управління базою даних система повинна підтримувати операції створення, читання, оновлення та видалення карток клієнтів, забезпечувати фільтрацію за сегментами і надавати інтерфейс для формування персоналізованих тригерних пропозицій.

Нефункціональні вимоги описують системні обмеження, архітектурні стандарти та параметри якості розробки. Бекенд-частина модуля має бути реалізована як асинхронний сервіс на базі сучасного веб-фреймворку під управлінням мови програмування високого рівня, тоді як фронтенд повинен базуватися на компонентній архітектурі та середовищі виконання коду на стороні сервера.

Основним транзакційним сховищем виступає об’єктно-реляційна база даних, яка підтримує посилальну цілісність.

Форматом обміну даними між клієнтською та серверною частинами виступає текстовий формат, а сам веб-інтерфейс адміністратора проєктується як кросбраузерний та адаптивний. При видаленні профілю користувача всі пов'язані логічні записи та поведінкові маркери мають знищуватися автоматично завдяки впровадженим механізмам каскадного видалення на рівні бази даних.

Постановка задачі розробки програмного модуля управління взаємодіями з клієнтами для меблевого інтернет-магазину Мебельки передбачає виконання низки взаємопов’язаних дослідницьких та інженерних завдань.

В кваліфікаційній (бакалаврській) роботі на основі виконаного аналізу поставлені такі завдання:

- спроєктувати структуру програмного модуля для меблевого магазину, бази даних для надійного збереження транзакційної та поведінкової інформації щодо взаємовідносин з клієнтами;
- реалізувати модуль збору даних, сегментації клієнтів та візуалізації статусу клієнтів;

- провести тестування розробленого програмного забезпечення.

Висновки до розділу 1

В першому розділі було здійснено дослідження предметної області та основних засад функціонування систем управління взаємовідносинами з клієнтами в умовах сучасної електронної комерції. Теоретичний аналіз дозволив встановити, що впровадження спеціалізованого програмного забезпечення є критичним фактором для забезпечення конкурентоспроможності онлайн-магазинів, оскільки воно дозволяє інтегрувати операційні та аналітичні процеси в єдину інформаційну екосистему.

Обгрунтовано виконано вибір технологій розробки, де пріоритетним визначено використання фреймворків. Такий підхід у поєднанні з реляційною моделлю збереження даних на базі прогресивних систем управління базами даних забезпечує необхідну відмовостійкість, цілісність інформації та високу швидкість обробки транзакційних запитів.

Проведений порівняльний аналіз існуючих аналогів на міжнародному та вітчизняному ринках виявив суттєву диспропорцію між функціональними можливостями великих корпоративних платформ та реальними потребами нішового малого бізнесу. Встановлено, що більшість універсальних рішень характеризуються надмірною складністю інтерфейсу, високою вартістю ліцензування та обмеженими можливостями для гнучкої адаптації аналітичних модулів під специфічні галузеві завдання, зокрема в сегменті меблевої торгівлі. Це підтверджує наявність вільної ринкової ніші для розробки програмного продукту, орієнтованого на інтелектуальну сегментацію клієнтської бази. Сформовані функціональні та нефункціональні вимоги, постановка задачі є основою для подальших досліджень та реалізації програмного модуля взаємовідносинами з клієнтами.

РОЗДІЛ 2
ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛЮВАННЯ ПРОЦЕДУР СЕГМЕНТАЦІЇ КЛІЄНТІВ CRM-СИСТЕМИ

2.1 Функціональна декомпозиція об’єкта розробки

Основним завданням системи є створення єдиного інформаційного середовища для автоматизації взаємодії з контрагентами, що потребує чіткого визначення ролей користувачів та їх прав доступу. Функціональна структура системи передбачає реалізацію модулів управління клієнтською базою, обробки багатокомпонентних замовлень, зберігання історії комунікацій та генерації аналітичних звітів. Нефункціональні вимоги охоплюють параметри продуктивності серверної частини, надійності збереження персональних даних та адаптивності інтерфейсу до різних типів пристроїв.

Для візуалізації логіки функціонування системи застосовується методологія об’єктно-орієнтованого аналізу. Побудова діаграми прецедентів дозволяє ідентифікувати основних акторів, таких як адміністратор системи та менеджер з продажів, а також визначити межі їхньої взаємодії з програмним продуктом. У межах проєктованої CRM-системи ключові сценарії включають реєстрацію нових звернень, ведення карток специфікацій меблевих виробів, моніторинг етапів виконання замовлення, формування вибірок для аналітичного модуля, формування сегментованих таблиць для визначення статусу клієнта [30-33].

На рисунку 2.1 представлена загальна структура програмного модуля взаємовідносин з клієнтом для меблевого магазину.

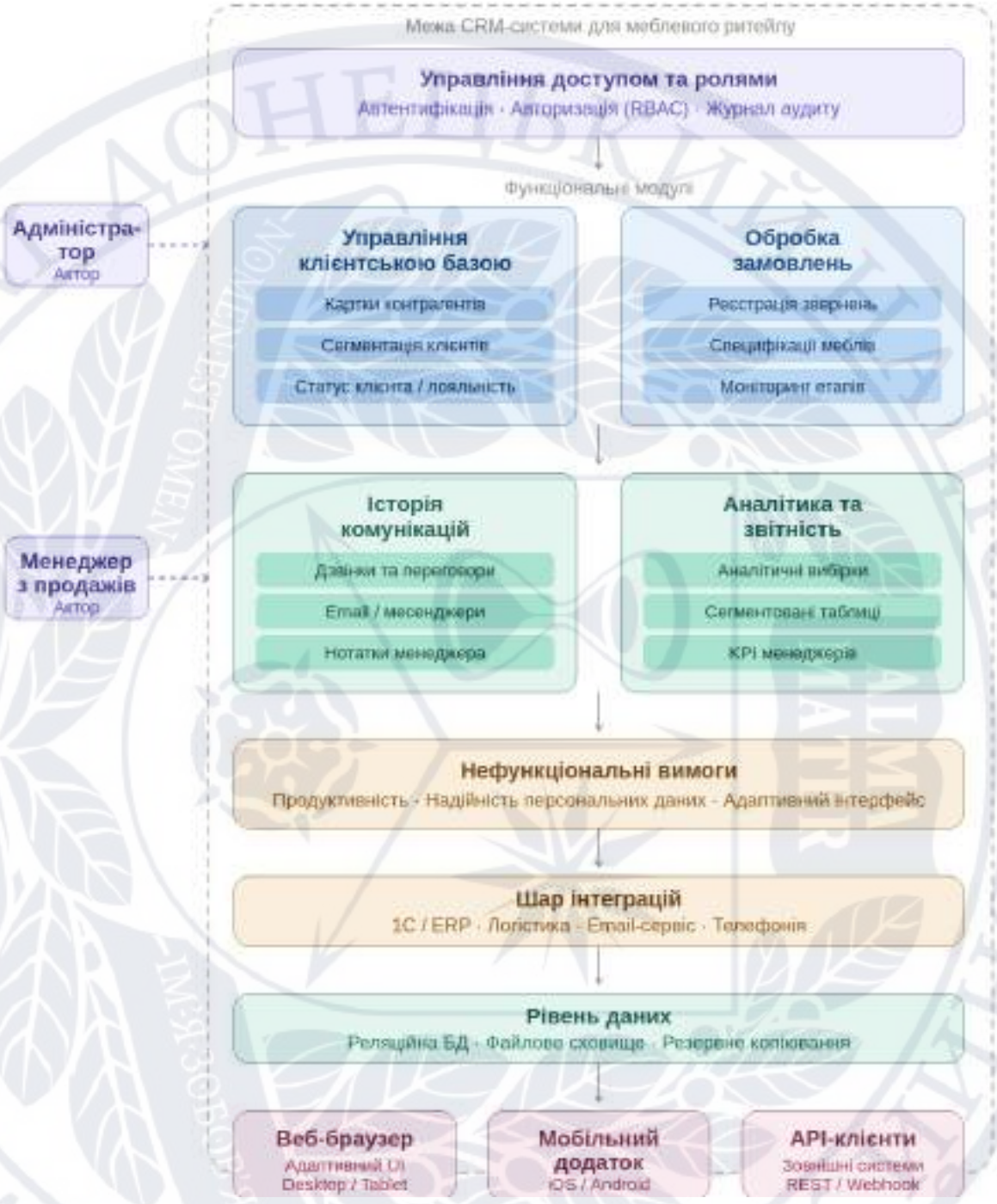

Рисунок 2.1 – Загальна структура системи взаємовідносин з клієнтом

Система передбачає використання алгоритму сегментації за методом RFM (Recency, Frequency, Monetary), який реалізується на рівні запитів до бази даних [32]. Для кожного клієнта розраховуються індивідуальні показники:

R – різниця між поточною датою та датою останнього замовлення,

F – загальна кількість успішних транзакцій,

M – сума всіх витрат.

Для визначення сегменту клієнтів, кожен параметр отримує свій бал, в результаті формуються спеціальні таблиці віп-клієнтів, стабільних клієнтів, клієнтів з одиночною закупкою, потенційних клієнтів, втрачених клієнтів.

Для поглибленого аналізу в систему інтегрується алгоритм кластеризації K-середніх. Процес починається з нормалізації даних, оскільки суми замовлень та їх кількість мають різні порядки величин. Після обробки алгоритм ітеративно визначає групи клієнтів, мінімізуючи цільову функцію J, яка визначається як сума квадратів помилок всередині кластера:

$$J = \sum_{\{i=1\}}^{\{k\}} \sum_{\{x \in S_i\}} \|x - \mu_i\|^2,$$

де k – кількість кластерів,

x – вектор ознак клієнта, який належить кластеру S_i,

μ_i – центр кластера S_i.

Таку функцію ще називають сумою квадратів помилок всередині кластера. Визначаючи квадрат Евклідової відстані між клієнтом та центром його сегмента відносно до середнього арифметичного всіх векторів x в даному сегменті, можна визначити кластери лояльних клієнтів, під загрозою відтоку тощо.

Це дозволяє виявити неявні групи, наприклад, «мисливці за знижками» або «покупці преміум-сегмента», що забезпечує автоматизацію маркетингових розсилок та персональні діалоги з клієнтами з точністю, яка недоступна при ручній обробці даних. Результати розрахунків зберігаються в окремому полі таблиці клієнтів, що дозволяє фронтенд-частині CRM миттєво відображати відповідні рекомендації для менеджера.

Результатом дослідження є загальна структура CRM-модуля, що базується на мікросервісному підході та ефективній схемі бази даних з використанням індексованих атрибутів і матеріалізованих представлень для прискорення аналітичних запитів. Впровадження розроблених алгоритмів автоматизованої сегментації дозволяє системі в реальному часі ідентифікувати пріоритетні групи клієнтів та адаптувати маркетингові стратегії під конкретні поведінкові сценарії.

Запропоноване програмне забезпечення забезпечує високу точність профілювання аудиторії та створює умови для масштабування бізнес-процесів онлайн-магазину. Використання комбінованого підходу до аналізу даних мінімізує похибки при прогнозуванні відтоку клієнтів та сприяє зростанню конверсії за рахунок релевантності персональних пропозицій.

2.2 Проєктування архітектури бази даних та інформаційного забезпечення

Інформаційне забезпечення системи будується на принципах реляційної моделі, що гарантує цілісність та несуперечливість даних при виконанні складних транзакцій. Процес проектування бази даних включає етапи концептуального, логічного та фізичного моделювання. На концептуальному рівні визначаються основні сутності, такі як клієнти, товари, замовлення, статуси та історія взаємодій. Особлива увага приділяється моделюванню зв’язків типу один до багатьох та багато до багатьох, що необхідно для коректного відображення структури складних замовлень, які можуть включати декілька одиниць меблевої продукції з унікальними характеристиками.

Важливою складовою аналітичного модуля проектованої системи є інструментарій для автоматизованої класифікації споживачів, що дозволяє диференціювати підходи до взаємодії з різними групами клієнтів. У межах розробки CRM-системи для нішового меблевого магазину обрано комбінований

підхід, що базується на методі RFM-аналізу та алгоритмі кластеризації К-середніх. Таке поєднання забезпечує як оцінку поточної цінності клієнта для бізнесу, так і виявлення прихованих поведінкових патернів, що не піддаються прямій логічній дескрипції.

Метод RFM-аналізу базується на дослідженні трьох ключових параметрів транзакційної активності: новизни останньої покупки, частоти замовлень за визначений період та сумарної грошової вартості всіх операцій. Математична суть методу полягає у присвоєнні кожному клієнту балів за кожним із критеріїв на основі квантильного розподілу всієї сукупності даних. Це дозволяє розділити базу на стійкі сегменти, такі як лояльні покупці, клієнти під загрозою відтоку та нові відвідувачі. Результати даного аналізу слугують вхідними даними для системи тригерних сповіщень та маркетингових розсилок, забезпечуючи високу точність таргетування при мінімальних обчислювальних витратах.

Для глибшого інтелектуального аналізу в системі реалізується алгоритм кластеризації К-середніх, який відноситься до методів навчання без учителя.

Алгоритм сегментації можна представити таким чином:

1. Визначення критеріїв (фінансові показники, категорія меблів, категорія приміщення тощо.
2. Визначення центроїду групи відповідно до показника.
3. Мінімізація сумарного квадратного відхилення об’єктів.

Основна ідея алгоритму полягає в ітераційному перерахунку центроїдів груп таким чином, щоб мінімізувати сумарне квадратне відхилення об’єктів усередині кожного кластера від його центру [33].

Програмна реалізація зазначених алгоритмів на серверній стороні дозволяє автоматично оновлювати сегменти при кожній новій транзакції. Це забезпечує динамічність системи та актуальність аналітичних звітів у реальному часі. Використання результатів кластеризації в інтерфейсі менеджера дозволяє персоналізувати процес консультування, спираючись на статистично обґрунтовані

прогнози потреб клієнта. Таким чином, інтеграція алгоритмів сегментації перетворює CRM-систему з інструменту реєстрації подій на інтелектуальну платформу підтримки прийняття управлінських рішень, що є важливим для підвищення ефективності нішового підприємства.

Логічна структура бази даних розробляється з урахуванням вимог нормалізації до третьої нормальної форми, що дозволяє мінімізувати дублювання інформації та уникнути аномалій при оновленні даних. Для реалізації фізичної моделі обрано систему управління базами даних PostgreSQL, яка дозволяє ефективно використовувати механізми індексування для прискорення пошукових запитів та забезпечує підтримку типів даних JSON для зберігання неструктурованих параметрів меблевих конфігурацій. Такий підхід забезпечує гнучкість системи при розширенні асортименту без необхідності повної перебудови схеми даних [16;17].

Процес проєктування оптимальної структури збереження даних передбачає мінімізацію надмірності інформаційного забезпечення, усунення аномалій модифікації, видалення та вставки, а також забезпечення логічної цілісності даних. Основним інструментом досягнення зазначених критеріїв є апарат нормалізації відносин, який полягає в послідовному переведенні вихідних структур даних до нормальних форм вищих порядків.

Для формалізації процесу нормалізації розглянемо предметну область «Облік реалізації продукції меблевого магазину». До базових сутностей системи належать: замовлення клієнтів, номенклатура меблевих виробів, контрагенти (постачальники) та клієнтська база.

1. Вихідний стан (Ненормалізована форма, 0НФ)

На етапі системного аналізу первинна інформація про транзакції часто консолідується в межах єдиного універсального відношення (таблиці), що містить складені атрибути та повторювані групи:

Замовлення_Ненормалізоване
= { (ID_Замовлення)¯, ПІБ_Клієнта, Телефон, Список_Товарів(ID_Товару, Назва, Категорія, Колір, Ціна),

Постачальник, Адреса_Постачальника}

Недоліки структури: наявність неатомарних значень (масивів даних у межах одного кортежу) унеможливлює коректне застосування апарату реляційної алгебри та стандартних засобів мови SQL.

2. Перша нормальна форма (1НФ)

Відношення перебуває в першій нормальній формі (1НФ) тоді і тільки тоді, коли всі його атрибути є атомарними (неподільними), а кожен кортеж містить лише одне значення для кожного з атрибутів.

Для приведення відношення до 1НФ здійснюється декомпозиція множинних атрибутів шляхом формування окремих рядків для кожної товарної позиції в межах замовлення.

Транзакції_1НФ
= { (*ID*_Замовлення)¯, (*ID*_Товару)¯, ПІБ_Клієнта, Телефон, Назва_Товару, Категорія, Колір, Ціна,

Постачальник, Адреса_Постачальника, Кількість}

Спостерігається надлишковість даних (дублювання ПІБ клієнта, телефонів, адрес постачальників при кожній транзакції). Виникає аномалія оновлення: зміна номера телефону клієнта потребує модифікації багатьох записів, що підвищує ризик втрати суперечності даних.

3. Друга нормальна форма (2НФ)

Відношення перебуває в другій нормальній формі (2НФ), якщо воно задовольняє вимогам 1НФ і кожен непервинний атрибут функціонально повно залежить від усього складеного первинного ключа, а не від його частини.

Шляхом проектування вихідного відношення на підмножини атрибутів здійснюється декомпозиція на три взаємопов’язані таблиці.

1. Замовлення. *ID_Замовлення,ПІБ_Клієнта,Телефон,Дата_Замовлення*

2. Товари.

$ID\_\text{Товару}, \text{Назва\_Товару}, \text{Категорія}, \text{Колір}, \text{Ціна}, \text{Постачальник}, \text{Адреса\_Постачальника}$

3. Специфікація замовлення. $ID\_\text{Замовлення}(FK), ID\_\text{Товару}(FK), \text{Кількість}$

У відношенні Товари наявна транзитивна функціональна залежність поля Товари від адреси постачальника. Це призводить до надмірності, якщо один постачальник забезпечує імпорт декількох категорій меблів.

4. Третя нормальна форма (3НФ)

Відношення перебуває в третій нормальній формі (3НФ), якщо воно відповідає критеріям 2НФ і жоден з неключових атрибутів не має транзитивної функціональної залежності від первинного ключа (усі неключові атрибути залежать безпосередньо і лише від первинного ключа).

Для усунення транзитивності з відношення Товари виокремлюється сутність «Постачальники». Додатково, з метою уніфікації та типізації текстових даних, доцільно винести атрибути «Клієнти» та «Категорії» в окремі довідникові відносини.

В результаті декомпозиції сформовано кінцеву логічну схему бази даних у 3НФ (Рисунок 2.2).

Рисунок 2.2 – Схема зв’язків

Розроблена в результаті нормалізації реляційна структура третьої нормальної форми (3НФ) може бути представлена у вигляді таких полів.

1. Постачальники. *ID*_Постачальника, Назва_Контрагента, Адреса, Телефон
2. Категорії меблів *ID*_Категорії, Назва_Категорії
3. Товари. *ID*_Товару , Назва, *ID*_Категорії, *ID*_Постачальника, Ціна, Колір
4. Клієнти. *ID*_Клієнта, ПІБ, Телефон, *Email*
5. Замовлення. *ID*_Замовлення, *ID*_Клієнта, Дата_Реєстрації
6. Вміст замовлення. *ID*_Замовлення, *ID*_Товару, Кількість

Сформована база даних дозволяє підтримувати логічну цілісність, мінімізувати дисковий простір та сформувати кортежі довідкових таблиць зі змінами для кожної транзакції.

2.3 Розробка архітектури програмного забезпечення та взаємодії компонентів

Архітектурне рішення системи базується на принципі розподілу відповідальності та використанні багаторівневої структури. Вибір мікросервісного або монолітного підходу з чітким розділенням фронтенд- та бекенд-частин зумовлений потребою в забезпеченні високої швидкості відгуку інтерфейсу. Бекенд-частина, реалізована на фреймворку API, функціонує як незалежний сервіс, що надає програмний інтерфейс для доступу до даних та виконання бізнес-логіки. Використання асинхронних драйверів для роботи з базою даних дозволяє системі одночасно обробляти значну кількість запитів без блокування основних процесів.

Клієнтська частина системи проектується як додаток на базі бібліотеки React.js. Архітектура фронтенду передбачає використання компонентного підходу, де кожен елемент інтерфейсу є незалежною одиницею з власним станом та

логікою відображення. Взаємодія між клієнтом та сервером реалізується за допомогою протоколу HTTP з використанням формату JSON для обміну даними. Для забезпечення безпеки передачі інформації передбачено використання механізмів автентифікації, що дозволяє надійно ідентифікувати користувачів та розмежовувати рівні доступу до конфіденційної інформації клієнтів [14; 30].

На рисунку 2.3 представлена загальна схема архітектури системи управління взаємовідносинами з клієнтами.

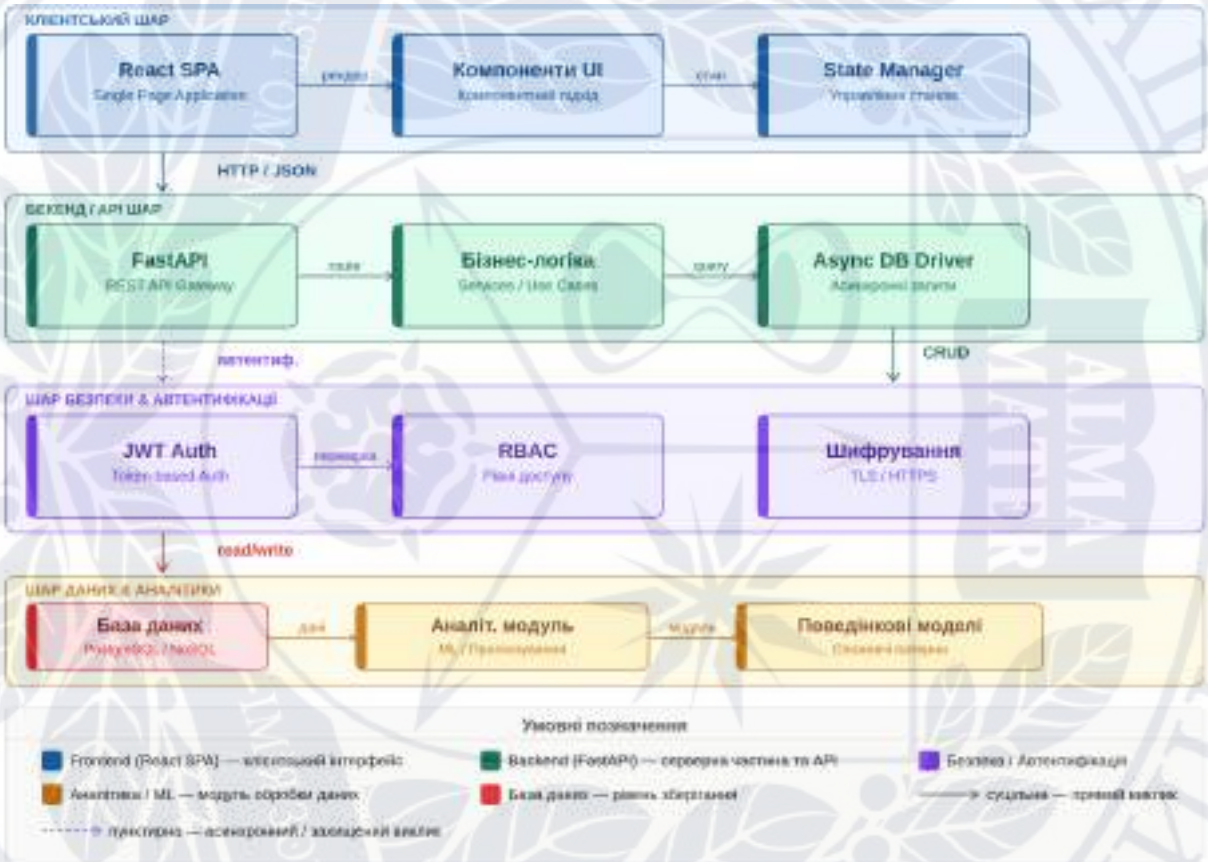

Рисунок 2.3 – Архітектура системи управління взаємовідносинами з клієнтами

На рисунку 2.4 представлена діаграма прецедентів для роботи інтернет-магазину та взаємодій з клієнтами. Діаграми UML дозволяють сформувати візуальні моделі для ефективної розробки [35].

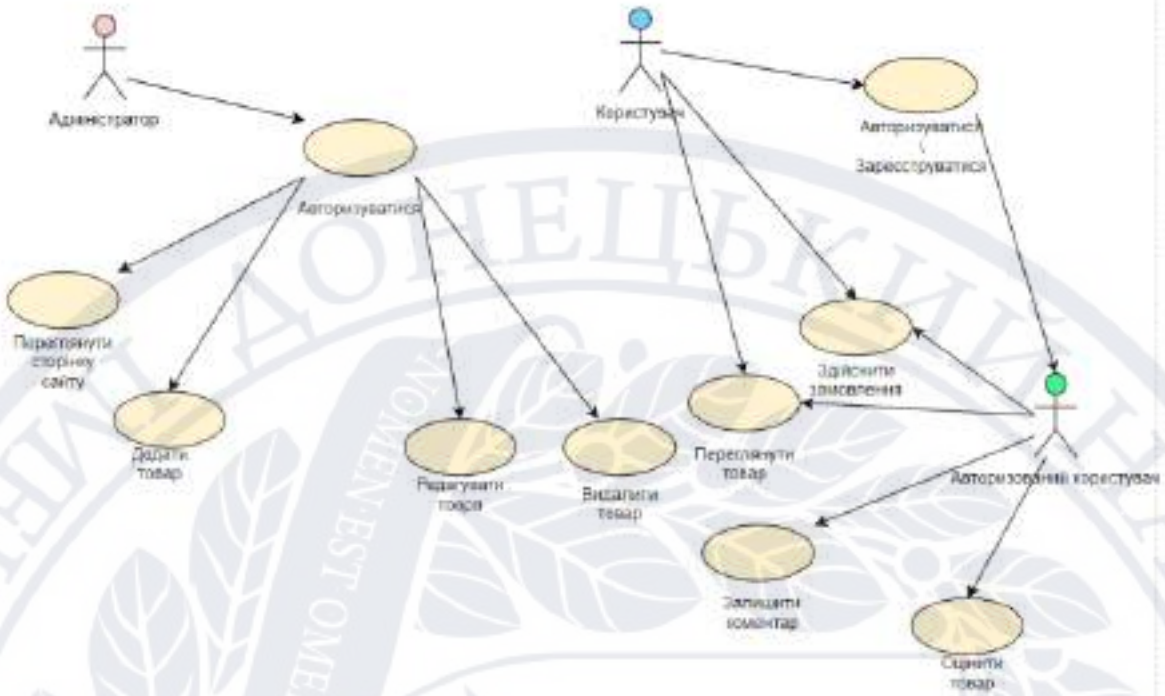

Рисунок 2.4 – UML–діаграма прецедентів роботи користувачів та адміністратора в інтернет-магазині

Для роботи з клієнтами необхідно сформувати загальну таблицю сегментації клієнтів для подальшої роботи з різними категоріями клієнтів (Таблиця 2.1).

Таблиця 2.1 Сегментація клієнтів (загальні характеристики)

Назва поля	**Тип даних**	**Опис**	**Приклад сегментації**
customer_id	PK (Int / UUID)	Унікальний ID клієнта	
first_name	Varchar	Ім’я	
email	Varchar	Електронна пошта	
phone	Varchar	Телефон	
registration_date	Date/DateTime	Дата реєстрації	За «віком» на сайті (Новачки vs Лояльні)
birth_date	Date	Дата народження	Вікові когорти (Gen Z, Міленіали тощо)
gender	Varchar/Enum	Стать (M, F, Unknown)	Чоловічі / Жіночі товари
country_city	Varchar	Геолокація	Географічні сегменти (Країна/Місто)

Така таблиця представляє первинні дані за полями ідентифікаційний номер, ім’я, реєстрація, день народження, гендер, країна/місто. Вона призначена для централізованого збереження антропометричних, реєстраційних та просторових характеристик користувачів, де як унікальний ідентифікатор використовується первинний ключ customer_id, а для маркетингової комунікації та авторизації застосовується унікальне поле e-mail. Ця таблиця є головним довідником суб’єктів, на який посилаються всі інші сегменти бази даних.

Транзакційні дані для сегментації представлені в таблиці 2.2.

Таблиця 2.2 Транзакційні дані для сегментації

Назва поля	**Тип даних**	**Опис**	**Приклад сегментації**
transaction_id	PK (Int / UUID)	ID транзакції	
customer_id	FK (Int / UUID)	Зв'язок з таблицею Customers	
order_date	DateTime	Дата та час покупки	Recency (Як давно купував)
total_amount	Decimal/Float	Сума чеку	Monetary (VIP-клієнти з високим чеком)
payment_status	Varchar/Enum	Статус (Completed, Refunded)	Категорія «Поверненці»
discount_used	Boolean	Чи використано промокод	«Мисливці за знижками» (Promo-driven)

Фінансова поведінка та комерційна активність покупців фіксуються у таблиці транзакцій. Таблиця містить поля первинного ключа transaction_id для кожної покупки та зовнішнього ключа customer_id, який реалізує пряме посилання на таблицю клієнтів (customers). Фінансові поля total_amount та хронологічні дані order_date дозволяють системі вираховувати давність і частоту замовлень для побудови фінансових моделей.

Паралельно з монетарними показниками, взаємодія користувачів із веб-ресурсом фіксується у таблиці поведінкової активності (Таблиця 2.3). Вона містить опис параметрів немонетарного залучення: ідентифікатор сесії activity_id, дату останнього візиту last_visit_date, кількість переглянутих сторінок

pages_viewed та маркер залишеного кошика abandoned_cart. Ця таблиця також містить зовнішній ключ customer_id, який забезпечує посилання на таблицю клієнтів (customers), пов’язуючи дії на сайті з конкретним профілем.

Таблиця 2.3 Поведінкові дані для сегментації

Назва поля	**Тип даних**	**Опис**	**Приклад сегментації**
activity_id	PK (Int / UUID)	ID дії	
customer_id	FK (Int / UUID)	ID клієнта (якщо авторизований)	
last_visit_date	DateTime	Дата останнього візиту	Активні / Відтік (Churn)
pages_viewed	Int	Кількість переглянутих сторінок	Залучені користувачі
favorite_category	Varchar	Категорія, яку дивились найбільше	Сегментація за інтересами (напр., "Скандинавський стиль")
abandoned_cart	Boolean	Чи залишив кошик	Сегмент «Кинуті кошики» для ремаркетингу

Класифікація та маркування споживачів здійснюється за допомогою довідника сегментів (segments), де зберігаються назви аналітичних груп segment_name та критерії відбору користувачів. В таблиці 2.4. є ключ segment_id та тип аналітики segment_type.

Оскільки один покупець може одночасно належати до кількох аналітичних категорій, зв’язок між ними реалізує проміжна асоціативна таблиця зв'язку (таблиця 2.5). Вона містить опис динаміки міграції клієнтів між категоріями за допомогою поля added_at та прапорця актуальності is_active.

Безпека та цілісність моделі забезпечується тим, що таблиця 2.5. не має власного простого ключа, а її первинний ключ є складеним і формується через одночасне посилання на таблицю клієнтів за допомогою зовнішнього ключа

customer_id та посилання на таблицю довідника сегментів через зовнішній ключ segment_id.

Таблиця 2.4 Довідник сегментів

Назва поля	**Тип даних**	**Опис**	**Приклад значень**
segment_id	PK (Int)	ID сегменту	1, 2, 2003
segment_name	Varchar	Назва сегменту	VIP_Покупці, Новачки, Сплячі
segment_type	Varchar/Enum	Тип (RFM, Демографія, Поведінка)	RFM
description	Text	Опис критеріїв сегменту	Купівля > 3 разів за останні 30 днів

При видаленні профілю користувача з головної таблиці customers, завдяки обмеженню, автоматично анулюються всі пов’язані посилання та записи в таблицях transactions, web_activity та customer_segments.

Таблиця 2.5 Зв’язок Клієнти та сегменти

Назва поля	**Тип даних**	**Опис**
customer_id	FK (Int) / Composite PK	Зв'язок з Customers
segment_id	FK (Int) / Composite PK	Зв'язок з Segments
added_at	DateTime	Коли клієнт потрапив у цей сегмент
is_active	Boolean	Чи актуальний сегмент для нього зараз

Відповідно до визначених таблиць та виконаних процедур нормалізації, була сформована ER схема зв’язків.

Концептуальна схема зв’язків розробленої бази даних відображає взаємодію між сутностями за принципом відносного підпорядкування, де основним ідентифікаційним ядром виступає таблиця клієнтів. Всі інші сутності системи прямо або опосередковано посилаються на цей центральний вузол, формуючи замкнену екосистему для всебічного аналізу споживчої поведінки (Рисунок 2.5.).

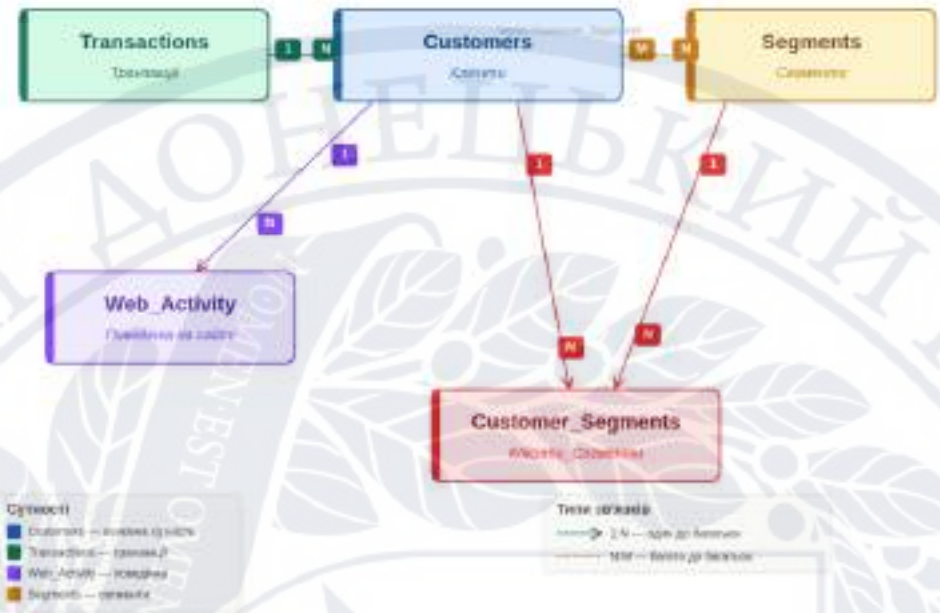

Рисунок 2.5 – Концептуальна схема зв’язків

Перший логічний контур зв’язків визначає монетарну та поведінкову активність користувачів. Відносини між таблицею клієнтів та таблицею транзакцій побудовані за класичною реляційною топологією один до багатьох (1:N). Це означає, що один унікальний запис суб’єкта з головної таблиці може асоціюватися з необмеженою кількістю фінансових операцій у підпорядкованій таблиці, або не мати їх взагалі, якщо користувач є новим лідом. Ідентифікація цього зв’язку реалізується через посилання зовнішнього ключа customer_id з таблиці замовлень на однойменний первинний ключ головної таблиці клієнтів. Аналогічний тип зв'язку один до багатьох (1:N) організовано між таблицею клієнтів та таблицею веб-активності, де кожна цифрова сесія чи лог взаємодії з меблевими категоріями на сайті прив'язується до конкретного користувача через зовнішній ідентифікатор customer_id.

Другий логічний контур реалізує архітектуру багатокритеріальної маркетингової сегментації, яка за своєю природою вимагає наявності зв'язку типу багато до багатьох (N:M) між таблицею клієнтів та довідником сегментів

(segments). Оскільки один покупець може одночасно володіти кількома аналітичними мітками, а кожен окремий сегмент містить у собі множину користувачів, пряме реляційне посилання між ними є неможливим. Для вирішення цієї задачі зв'язок декомпоновано на два незалежні відношення типу один до багатьох (1:N) за допомогою проміжної асоціативної таблиці (customer_segments).

Напрямок цих зв'язків концентрується всередині асоціативного вузла: таблиця клієнтів посилається один до багатьох на таблицю зв'язку через зовнішній ключ customer_id, а таблиця довідника сегментів аналогічно посилається один до багатьох на цей же вузол через зовнішній ключ segment_id. Разом ці два посилання утворюють складений первинний ключ асоціативної таблиці, що повністю виключає дублювання однакових статусів для одного користувача.

Цілісність та каскадність усієї концептуальної схеми забезпечується жорсткими обмеженнями на рівні бази даних. Усі зовнішні ключі, що беруть свій початок із головної таблиці клієнтів спрямовані до таблиць транзакцій, веб-активності (web_activity) та зв'язку із сегментами (customer_segments), забезпечені правилом, що у разі анулювання профілю клієнта інформаційна система автоматично і миттєво знищить усі пов'язані посилання, очищуючи фінансову історію, логи сайту та маркетингові мітки без порушення реляційної структури всієї бази даних.

На рисунку 2.6 представлена діаграма послідовностей, яка демонструє запит користувача.

Користувач реєструється в системі і в подальшому заходить через свій логін. Він може оглядати сайт, здійснювати покупки, аналізувати свої попередні покупки, аналізувати пропозиції та знижки, спілкуватись з менеджером, користуватись візуальними аналітичними матеріалами.

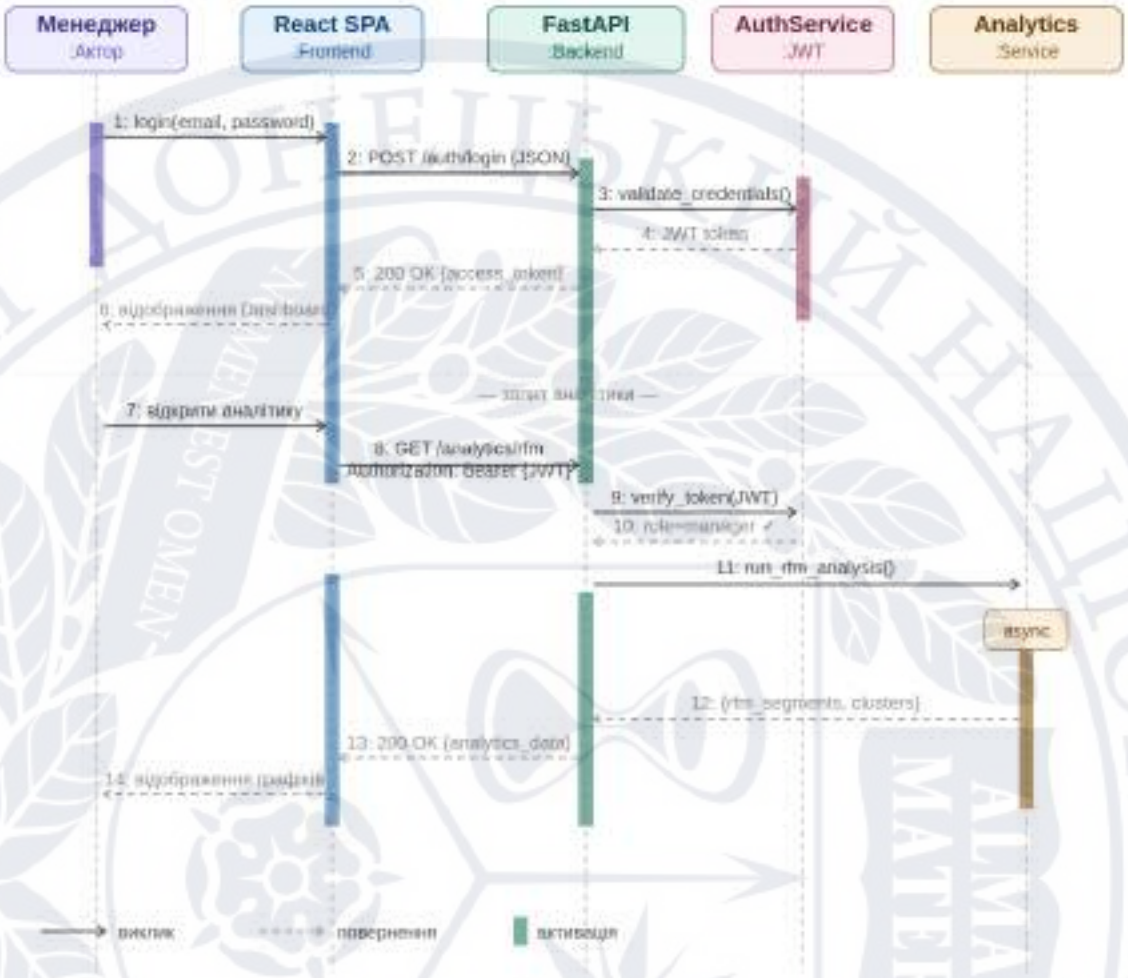

Рисунок 2.6 – Діаграма послідовностей запиту користувача

В подальшому буде реалізовано програмний модуль сегментації клієнтів.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи виконано комплекс проектування архітектурних рішень та моделювання ключових бізнес-процесів інтелектуальної CRM-системи, адаптованої до специфіки функціонування підприємств сфери нішового меблевого ритейлу.

За допомогою побудови уніфікованих UML-діаграм прецедентів ідентифіковано ключових акторів (адміністратора та менеджера з продажів) та декомпоновано базові сценарії їхньої інтеракції з програмним комплексом, що дозволило чітко розмежувати рівні доступу та оптимізувати операційні алгоритми обробки замовлень і ведення специфікацій меблевої продукції.

Спроектовано дворівневу математичну модель інтелектуального аналізу клієнтської бази, яка комбінує детермінований RFM-аналіз) на базі квантильного розподілу транзакційної активності та стохастичний алгоритм кластеризації К-середніх із мінімізацією цільової функції суми квадратів помилок усередині кластерів на основі Евклідової відстані.

Здійснено послідовну нормалізацію інформаційного забезпечення від вихідного стану до третьої нормальної форми, що дозволило декомпозувати складні неатомарні структури даних меблевих замовлень на систему взаємопов’язаних таблиць постачальників, категорій, товарів, клієнтів, замовлень та їхнього вмісту. Розроблено фізичну архітектуру бази даних у середовищі СУБД PostgreSQL

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ CRM-СИСТЕМИ

3.1 Особливості роботи сайту та сторінок соціальних мереж онлайн магазину «Мебельки»

Аналіз специфіки функціонування та архітектури цифрових каналів взаємодії інтернет-магазину меблів «Мебельки» дозволяє виділити комплекс технологічних, організаційних та маркетингових особливостей. Ефективність сучасного нішевого ритейлу безпосередньо залежить від синергії між офіційним веб-ресурсом компанії та її представництвами у соціальних медіа, які разом формують єдину омніканальну екосистему для оптимізації взаємодії з кінцевим споживачем [36].

Веб-платформа бренду проєктується як інтерактивний інструмент підтримки прийняття рішень, що враховує високий середній чек і тривалий цикл ухвалення рішення про покупку меблевої продукції. На відміну від ритейлу товарів повсякденного попиту, меблі вимагають значної варіативності конфігурації, включаючи вибір типу тканини, матеріалу корпусу, фурнітури та габаритів. Сайт реалізує інтерактивні модулі моделювання, де кожна зміна властивості динамічно перераховує вартість виробу. На рівні збереження даних це забезпечується використанням об’єктів неструктурованих параметрів у системі управління базами даних, що дозволяє клієнтській частині миттєво моніторити зміни без перебудови архітектури таблиць. Важливою особливістю сайту є фіксація немонетарної активності користувачів, де система відстежує глибину залученості, тривалість перебування в конкретних категоріях та фіксує тригерні події, такі як залишений кошик. Будь-яка дія на сайті миттєво передається через програмний інтерфейс і відображається на панелі менеджера, виступаючи первинним джерелом для

наповнення таблиць транзакцій та логів активності, які є базисом для аналітичних алгоритмів сегментації.

Соціальні платформи інтернет-магазину виконують роль первинного контуру залучення холодного трафіку, візуальної презентації естетики інтер'єрів та інструменту побудови довгострокової лояльності. Меблевий ритейл залежить від візуального сприйняття, тому офіційні сторінки в соціальних мережах орієнтовані на демонстрацію готових інтер’єрних рішень, відеоогляди механізмів трансформації та тестів зносостійкості матеріалів, що дозволяє нівелювати бар’єр дистанційної покупки. Сторінки платформ функціонують як інтегровані торгові майданчики за допомогою вбудованих інструментів каталогів. Користувач має можливість ініціювати діалог або замовити консультацію безпосередньо через месенджери, що потребує інтеграції соціальних інтерфейсів із внутрішньою CRM-системою для автоматичного створення картки потенційного клієнта. Взаємодія аудиторії зі сторінками у вигляді вподобань, збережень публікацій чи коментарів дозволяє збагачувати маркетингові профілі користувачів. Використання пікселів відстеження допомагає пов’язати профілі в соціальних мережах із діями на веб-ресурсі, створюючи умови для точного динамічного ремаркетингу.

Головною особливістю архітектури є безшовний перехід клієнта між цифровими точками дотику в межах єдиної стратегії. Користувач, залучений через візуальний контент у соціальній мережі, переходить на сайт для індивідуального конфігурування меблів. Якщо сесію перервано на етапі кошика, інтегрована CRM-система ініціює автоматизований сценарій, при якому на сайті фіксується статус гарячого ліда, а клієнт отримує персоналізовану тригерну пропозицію або наздоганяючу рекламу саме тих моделей меблів, якими він цікавився. Такий підхід забезпечує максимальну конверсію та оптимізує витрати на залучення покупців у нішевому сегменті ритейлу. Для реалізації програмного модуля будемо використовувати вебтехнології [37].

3.2 Реалізація серверної частини та структури бази даних у середовищі Node.js

Процес програмної реалізації CRM-системи розпочато з розгортання серверної інфраструктури за допомогою Node.js. Вибір фреймворку дозволив використати подійно-орієнтовану модель для забезпечення асинхронної взаємодії між компонентами системи. Основна логіка обробки запитів зосереджена в контролерах, які взаємодіють з базою даних PostgreSQL через об’єктно-реляційне відображення. Це забезпечило високу швидкість розробки та легкість підтримки коду при маніпуляціях зі складними структурами даних. Схема бази даних була імплементована з урахуванням необхідності зберігання розширених профілів клієнтів, деталізованих специфікацій меблевих виробів та логів взаємодії, що вимагало створення набору зв’язаних таблиць із чітким контролем цілісності на рівні зовнішніх ключів. Реалізація програмного модуля буде виконана за допомогою JavaScript, бібліотек та фреймворків [38; 39].

Особлива увага приділена розробці прикладного програмного інтерфейсу за стандартом REST. Серверна частина реалізує набір кінцевих точок для виконання операцій створення, читання, оновлення та видалення даних. Для захисту конфіденційної інформації впроваджено механізми автентифікації та авторизації на основі веб-токенів, що дозволяє розмежовувати доступ до клієнтської бази між різними категоріями персоналу магазину. Асинхронна природа Node.js у поєднанні з оптимізованими запитами до PostgreSQL забезпечила мінімальний час відгуку системи навіть при одночасній обробці великих масивів аналітичної інформації.

Серверна частина застосунку організована за модульним принципом і розподілена на контролери бізнес-логіки та маршрутизатори запитів. Модуль конфігурації підключення до бази даних створює пул з’єднань із СУБД PostgreSQL, що оптимізує роботу за умов конкурентних запитів. Маршрутизація

чітко розмежовує операції над сутностями системи, розподіляючи їх на потоки обробки замовлень, автентифікації користувачів, управління товарами та формування аналітики. Фрагменти реалізованого коду представлені в додатку А.

3.3 Розробка клієнтського інтерфейсу на базі бібліотеки React.js

Клієнтська частина системи забезпечує моніторинг безперервного досвіду користувача без необхідності повного перезавантаження сторінок інтерфейсу. Використання бібліотеки React.js дозволило побудувати модульну архітектуру, де кожна панель управління, таблиця замовлень або графік аналітики є незалежним компонентом. Стан застосунку контролюється за допомогою інструментів централізованого управління потоками даних, що гарантує синхронізацію відображуваної інформації на всіх екранах системи при внесенні змін менеджером або автоматичному оновленні даних із зовнішніх джерел [10].

Інтерфейс системи адаптований до потреб нішового меблевого ритейлу та містить спеціалізовані форми для конфігурування параметрів виробів, таких як розміри, матеріали та фурнітура. Візуалізація результатів сегментації та кластеризації клієнтів реалізована за допомогою динамічних діаграм, що отримують дані від аналітичного модуля в реальному часі. Висока продуктивність фронтенд-частини досягнута завдяки оптимізації моніторингу компонентів та використанню лінивого завантаження окремих модулів системи, що є важливим для стабільної роботи адміністративної панелі в браузері.

Клієнтська частина застосунку відповідає за формування динамічного інтерфейсу адміністратора та менеджерів з продажу. Компонент аналітичної панелі реалізує первинне завантаження статистичних даних про загальну кількість клієнтів, обсяг замовлень та сукупний дохід через HTTP-запити до серверних ендпоінтів за допомогою бібліотеки Axios. Отримані дані динамічно візуалізуються у вигляді інформаційних карток та інтерактивної таблиці останніх

замовлень із відображенням їхніх поточних статусів. Окремі інтерфейсні модулі забезпечують відображення структурованого каталогу меблевої продукції з можливістю додавання нових позицій через спеціалізовані екранні форми, а також інтерфейс управління користувачами, де реалізовано механізм перегляду розширеної аналітики щодо купівельної спроможності та активності кожного покупця.

Основна аналітична логіка зосереджена в контролерах, які взаємодіють із базою даних за допомогою прямих SQL-запитів. Для формування загальної статистики панелі продавця реалізовано агрегаційні запити з використанням математичних функцій підрахунку кількості та обчислення суми транзакцій, а також обробку потенційних порожніх значень за допомогою оператора заміни NULL на нульовий показник. Особливе місце посідає алгоритм класифікації та сегментації клієнтської бази, інтегрований у серверний шар. Він аналізує історію замовлень кожного користувача, обчислюючи ключові поведінкові метрики: давність здійснення останньої покупки, частоту купівель та загальну суму витрачених коштів. На основі цих детермінованих маркерів система автоматично категоризує покупців за рівнем їхньої лояльності та залученості. Фрагменти реалізованого коду представлені в додатку А.

Для синхронізації з сайтом Мебельки доцільно використовувати спеціальні модулі щодо імпорту даних, використання аналітики, експорт даних з сайту після змін. Це дозволило акумулювати всю історію покупок та переглядів товарів у єдиному сховищі CRM-системи. Використання офіційних API соціальних мереж Facebook та Instagram. цих платформ дозволяє системі збирати агреговану інформацію про активність аудиторії, ефективність рекламних кампаній та прямі запити користувачів у месенджерах. Отримані дані інтегруються в профіль клієнта, створюючи багатовимірну модель його поведінки. Такий підхід забезпечує аналітичному модулю доступ до зовнішніх сигналів інтересу споживачів, що значно підвищило точність алгоритмів кластеризації та дозволило

менеджерам магазину формувати персоналізовані пропозиції не лише на основі минулих покупок, а й на основі активності клієнтів у соціальному медіапросторі. У результаті програмний продукт є комплексним інструментом підтримки прийняття рішень, що об'єднує внутрішню операційну логіку з потужними зовнішніми потоками даних. Представлений програмний модуль в подальшому може бути інтегрований з різними профілями онлайн магазину «Мебельки».

3.4 Тестування програмного модуля управління взаємовідносинами з клієнтами

Тестування програмного модуля управління взаємовідносинами з клієнтами інтернет-магазину меблів Mebelki є важливим етапом розробки, спрямованим на верифікацію функціональної придатності, надійності, продуктивності та цілісності даних розробленої системи. Враховуючи інтеграцію алгоритмів інтелектуального аналізу даних та автоматизованої сегментації, процес тестування охоплює комплекс методів, що перевіряють роботу системи на різних рівнях абстракції [40].

Функціональне тестування проводиться для перевірки відповідності реалізованого програмного забезпечення сформованим функціональним вимогам. Об'єктом верифікації виступають інтерфейси користувачів та сценарії взаємодії. Тестуванню підлягають модулі реєстрації та автентифікації користувачів, створення та модифікації карток клієнтів, ведення специфікацій меблевої продукції з урахуванням динамічної зміни атрибутів, а також процеси обробки багатокомпонентних замовлень.

Для систем управління взаємовідносинами з клієнтами доцільно перевірити процедури зміни статусів клієнтів на основі досвіду транзакцій.

Інтеграційне тестування забезпечує перевірку взаємодії між ізольованими компонентами системи. У межах розробленої архітектури аналізується безшовний

обмін даними між клієнтським додатком на React.js та серверною частиною на API через протокол HTTP у форматі JSON. Перевірці підлягає надійність механізмів автентифікації на основі токенів при передачі конфіденційної інформації та коректність асинхронного виконання пошукових запитів. Компонентне тестування підтверджує ізольовану працездатність окремих сервісів і матеріалізованих представлень для прискорення аналітичних обчислень.

Нефункціональне тестування оцінює параметри продуктивності та захищеності системи.

Навантажувальне тестування дозволяє визначити межі стабільної роботи серверної частини при одночасному опрацюванні значної кількості транзакцій та аналітичних запитів.

Тестування безпеки фокусується на перевірці стійкості системи до несанкціонованого доступу до персональних даних клієнтів меблевого магазину. Успішне проходження всіх етапів тестування підтверджує готовність програмного модуля до розгортання та промислової експлуатації.

В межах виконання бакалаврської роботи було виконано функціональне тестування. кластерів та стійкість формування неявних поведінкових груп.

Тестування архітектури бази даних та інформаційного забезпечення спрямоване на контроль посилальної цілісності та відсутності аномалій.

За допомогою спеціально згенерованих тестових сценаріїв перевіряється робота реляційної структури третьої нормальної форми (3НФ).

Тестування тригерів реального часу, які відповідають за автоматичну детекцію гарячих лідів та реактивацію сплячих клієнтів при виявленні залишених кошиків, проводиться шляхом імітації поведінкових логів у таблиці веб-активності.

Важливим елементом є верифікація обмежень каскадного оперування для підтвердження автоматичного та коректного видалення залежних фінансових і маркетингових міток при анулюванні первинних записів користувачів.

В межах кваліфікаційної бакалаврської роботи було виконано функціональне тестування за визначеними кейсовими процедурами.

На рисунку 3.1 представлено вид панелі продавця, який додає товар (процедура додавання товару).

Продавець бачить кількість клієнтів, замовлень, сумарний дохід, додає товари. Важливим є інтеграція профілів продавців, якщо їх багато. Як правило, на фірмі розділяють бази клієнтів або категорії меблів. Представлений програмний модуль тестувався для одного продавця.

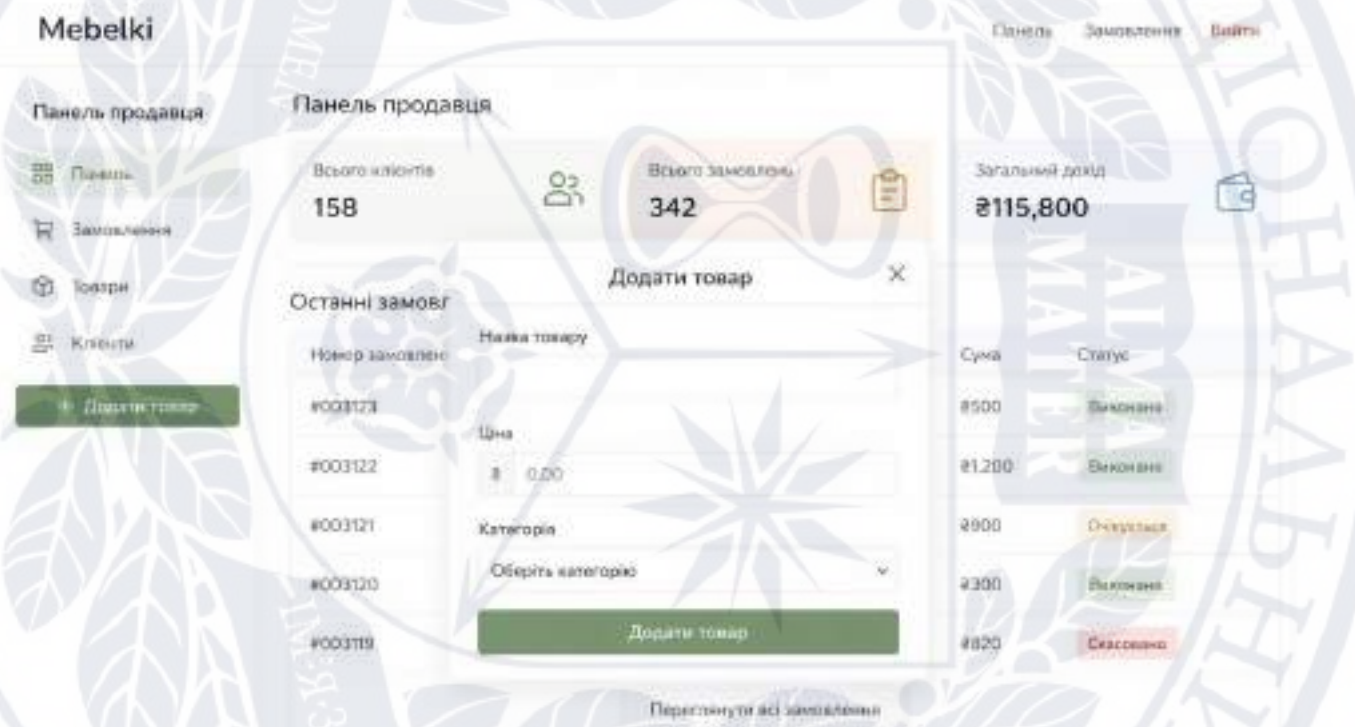

Рисунок 3.1 – Панель продавця

На рисунку 3.2 представлена спрощена панель сегментації клієнтів (процедура сегментації).

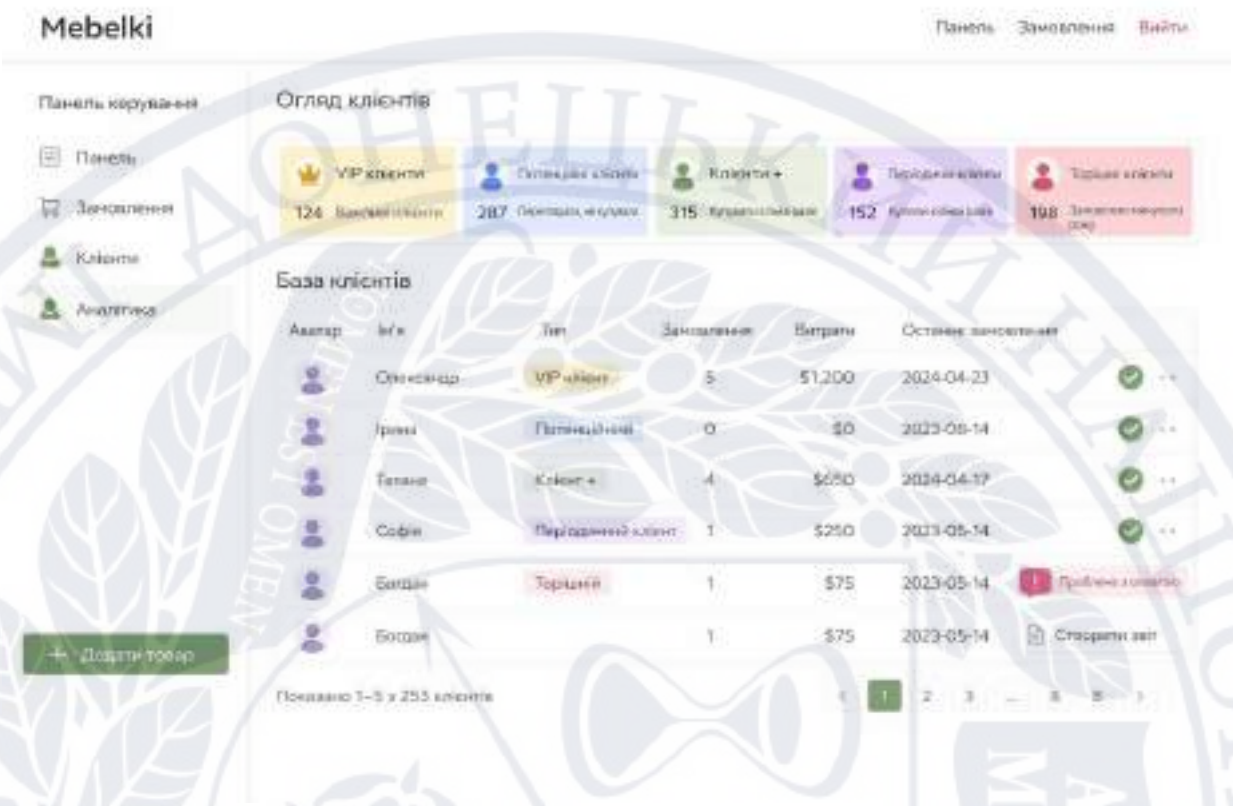

Рисунок 3.2 – Панель сегментації клієнтів

Функціонально були перевірені процедури реєстрації та формування профілю клієнта (Рисунок 3.3 – процедури реєстрації та формування профілю).

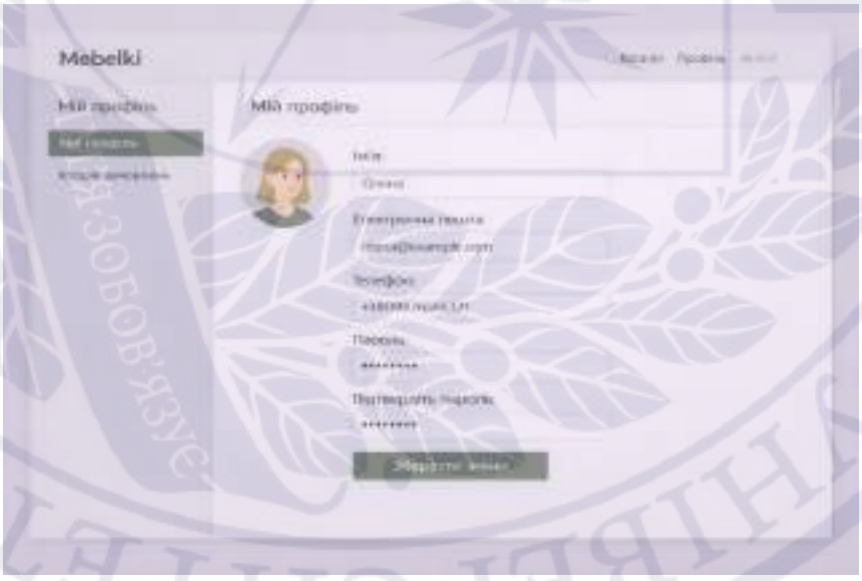

Рисунок 3.3 – Панель профілю клієнта

Також було перевірено формування таблиць замовлень (Рисунок 3.4 – процедура формування замовлень).

Рисунок 3.4 – Панель «Замовлення»

Розглянемо процедуру автоматичної модифікації маркетингового сегменту клієнта при фіксації тригерної поведінки на веб-ресурсі.

На рисунку 3.5 представлено процедуру тестування за кейсом – Список клієнтів з відміткою проблеми з оплатою.

У базі даних сформовано мета-довідник сегментів, який містить аналітичні категорії привабливий клієнт, періодичний, торішній, потенційний. У таблиці клієнтів присутній зареєстрований користувач із фіксованим унікальним ідентифікатором. У таблиці транзакцій для зазначеного контрагента наявні записи про успішні операції, здійснені до моменту тестування. У таблиці поведінкової активності записи для цього клієнта відсутні. Збережені процедури розрахунку метрик перебувають в активному стані.

Рисунок 3.5 – Список клієнтів, моніторинг дій

Після того, як здійснюється виклик збереженої процедури аналізу з наступним виконанням аналітичного запиту до асоціативної таблиці сегментації клієнтів для перевірки поточного активного статусу суб'єкта, результатом є присвоєння статусу і зміни в таблицях.

На рисунку 3.6 представлено перехід клієнта з одної категорії в іншу. Проводиться емуляція повернення користувача на сайт та формування купівельного наміру. Виконується операція вставки кортежу в таблицю активностей з поточною реєстрацією часу, кількості переглянутих сторінок, виконаної покупки. Автоматично спрацьовує тригер після вставки рядка в таблицю

активностей.

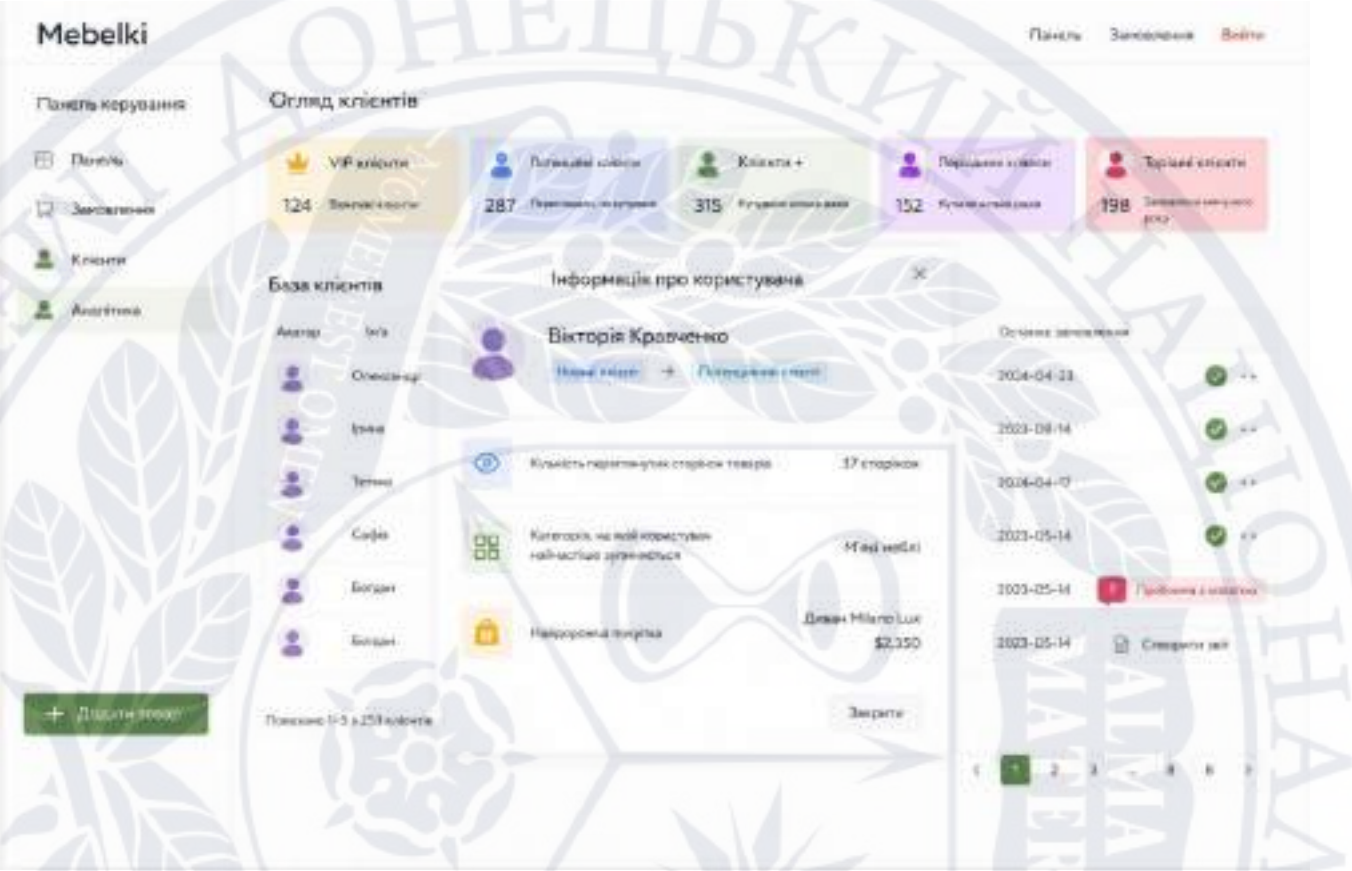

Рисунок 3.6 – Зміна категорії клієнта

У таблиці генерується новий активний запис із посиланням на виконану транзакцію. Якщо всі процедури виконані, то клієнт з потенційного перетворюється на періодичного або привабливого, якщо фінансові розрахунки відповідають рівню.

Результати функціонального тестування представлені в таблиці 3.1.

Аналіз наведених у таблиці даних свідчить про повну працездатність усіх спроєктованих модулів та інтерфейсів інформаційної системи. Під час проведення функціонального контролю методом чорної скриньки було підтверджено безпомилковість реєстрації користувачів, створення карток клієнтів із генерацією унікальних ідентифікаторів у системі управління базами даних PostgreSQL, а

також коректність динамічного відображення замовлень, товарних специфікацій та сумарних фінансових показників на аналітичній панелі продавця.

Таблиця 3.1 Результати функціонального тестування

№ процедури	Процедура тестування	Метод тестування	Очікуваний результат	Фактичний результат
1	Реєстрації та формування профілю клієнта	Функціональний контроль	Коректне внесення первинних антропометричних даних, створення картки користувача	Профіль успішно сформовано, дані занесені до відповідних полів СУБД PostgreSQL
2	Додавання товару та ведення специфікацій	Перевірка бізнес-логіки інтерфейсу	Можливість додавання нових позицій меблів продавцем,	Товарні позиції додаються, базова інформація коректно оновлюється
3	Формування замовлень на панелі продавця	Емуляція заповнення форм замовлення	Створення таблиць замовлень, відображення показників	Фінансові та кількісні показники замовлень успішно відображаються в інтерфейсі
4	Відображення спрощеної панелі сегментації	Візуальний контроль виведення списків	Виведення розрахованих аналітичних груп на екран	Списки категорій користувачів успішно згенеровані та виведені на панель
5	Автоматична модифікація маркетингового сегменту	Автоматизована перевірка тригерної логіки	Спрацьовування тригера автоматичне переведення зі статусу	Тригер успішно активується після фіксації сесії, у таблиці генерується новий запис

Емуляція поведінкових чинників та фіксація користувацьких сесій підтвердили здатність системи оперативно реагувати на тригерні події, змінювати статус суб’єктів у реальному часі та забезпечувати реляційну цілісність даних.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи виконано комплекс робіт із програмної реалізації, інтеграції та верифікації CRM-системи інтернет-магазину меблів Mebelki.

Досліджено технологічні та комунікативні особливості цифрової інфраструктури об’єкта розробки, що об’єднує інтерактивний веб-ресурс та офіційні сторінки в соціальних мережах Facebook та Instagram в єдину омніканальну екосистему.

Реалізовано асинхронну серверну архітектуру в середовищі Node.js із використанням об’єктно-реляційного відображення для взаємодії з базою даних PostgreSQL.

На основі компонентного підходу бібліотеки React.js розроблено клієнтську частину CRM-системи.

Виконано функціональне тестування програмного модуля на різних рівнях абстракції, включаючи верифікацію інтерфейсів додавання товарів, карток замовлень та профілів користувачів.

ВИСНОВКИ

У бакалаврській роботі проведені теоретико-практичні дослідження, спрямовані на розробку й верифікацію програмного модуля управління взаємовідносинами з клієнтами для нішевого інтернет-магазину меблів «Мебельки».

На основі детального аналізу теоретичних засад функціонування сучасних систем автоматизації комерційної діяльності у сегменті електронної комерції визначено особливості та кейси використання прикладного програмного забезпечення для нішевого ритейлу. Установлено, що специфіка меблевого ринку вимагає гнучких інструментів для обробки тривалого циклу прийняття рішень про покупку та інтерактивного конфігурування багатокомпонентних специфікацій виробів. Дослідження наявних архітектурних рішень показало необхідність переходу від ізольованих систем реєстрації подій до створення цілісних омніканальних екосистем, здатних об’єднувати інформаційні потоки офіційного сайту підприємства та його представництв у соціальних мережах для підвищення конверсії та оптимізації витрат на залучення покупців.

Шляхом вивчення сучасних технологій веб-розробки та інструментів інтелектуального аналізу даних обґрунтовано вибір технологічного стеку для реалізації проєкту. Платформа Node.js забезпечила побудову асинхронної подійно-орієнтованої серверної архітектури, здатної ефективно взаємодіяти з реляційним сховищем PostgreSQL та прискорювати аналітичні розрахунки за рахунок індексування атрибутів і матеріалізованих представлень. Для створення інтерфейсу користувача обрано бібліотеку React.js, яка на основі компонентного підходу дозволила реалізувати застосунок із високою швидкістю відгуку та можливістю динамічної візуалізації аналітичних даних.

У межах формування функціональної декомпозиції об’єкта розробки формалізовано перелік вимог до CRM-системи, побудовано UML-діаграми та архітектурні схеми міжкомпонентної взаємодії.

Результатом декомпозиції складених атрибутів замовлень стала логічна схема бази даних, що складається з взаємопов’язаних сутностей клієнтів, транзакцій, веб-активності та сегментів, захищених декларативними обмеженнями каскадного видалення даних.

З метою автоматизації процесів взаємодії та точного профілювання аудиторії розроблено прототип програмного модуля з інтегрованими алгоритмами сегментації клієнтів. Система поєднує детермінований RFM-аналіз для розрахунку давності, частоти та монетарної вартості успішних транзакцій із алгоритмом кластеризації K-середніх на основі нормалізованих векторів ознак та Евклідової відстані.

Виконано функціональне тестування розробленого програмного модуля, яке підтвердило його повну працездатність, надійність та відповідність сформованим технічним вимогам. Перевірка логіки автоматичної зміни сегментів на основі розроблених тестових сценаріїв та кейсів продемонструвала безпомилкове функціонування тригерів реального часу на рівні бази даних та збереження посилальної цілісності при модифікації записів.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Краус К.М., Краус Н.М., Манжура Н.В. Електронна комерція та інтернет-торгівля. Навчально-методичний посібник. Київ: Аграр-медіа груп, 2021. 454 с.

2. A Comprehensive Guide to the CRM Selection Process. *3andfour.* URL: https://www.3andfour.com/articles/crm-selection-process. (дата звернення: 20.05.2026).

3. Your definitive CRM selection guide and checklist. *Discovercrm*. URL: https://www.discovercrm.com/crm-selection-guide-andchecklist.html. (дата звернення: 20.05.2026).

4. Фриз І.В., Гуцуляк Д.С. Програмний модуль управління взаємовідносинами з клієнтамионлайн магазину електронної комерції. Матеріали Всекраїнської науково-практичної конференції здобувачів вищої освіти та молодих вчених «Прикладні інформаційні технології», 15 травня 2026 р., Донецький національний університет імені Василя Стуса, м. Вінниця. 2026.

5. How to Select the Best CRM. *KAIZEN™*. URL: https://kaizen.com/insights/crm-selection/. (дата звернення: 20.05.2026).

6. Basic Criteria for Choosing a CRM System. *Ascendix*. URL: https://ascendix.com/blog/crm-selection-process/. (дата звернення: 20.05.2026).

7. 8 tips for choosing the right CRM for your business. *Superoffice*. URL: https://www.superoffice.co.uk/resources/articles/choose-crmvendor/. (дата звернення: 20.05.2026).

8. 10 найкращих CRM-систем в Україні. *Crmsolutions*. URL: https://crmsolutions.ua/top-10-best-ukrainian-crm-systems/. (дата звернення: 20.05.2026).

9. Савран, Н. В. CRM-Система: Етапи розвитку та класифікація видів. *Економічний простір*, 2021. 168. С. 72-77.

10. FastAPI framework, high performance, easy to learn, fast to code, ready for production. *FastAPI*. URL: https://fastapi.tiangolo.com/. (дата звернення: 10.05.2026).

11. Весьолов В. Впровадження CRM системи – повна інструкція. *Sendpulse*. URL:https://sendpulse.ua/blog/srm-system-implementation. (дата звернення: 20.05.2026).

12. Node.js JavaScript runtime built on Chrome's V8 JavaScript engine. URL: ю (дата звернення: 15.05.2026).

13. Рейтинг фреймворків для веброзробки. Btander. URL: https://brander.ua/blog/reytynh-freymvorkiv-dlya-vebrozrobky. (дата звернення: 10.05.2026).

14. React – A JavaScript library for building user interfaces. URL: https://react.dev/ . (дата звернення: 20.05.2026).

15. Вівчарик В. Повний огляд REST: нюанси, поради, приклади. *DOU*. URL: https://dou.ua/forums/topic/50364/. (дата звернення: 20.05.2026).

16. Павловський І. В., Петрашенко А. В. Бази даних та засоби управління: підручник. Київ: КПІ ім. Ігоря Сікорського, 2024. 293 с.

17. PostgreSQL: The World's Most Advanced Open Source Relational Database. URL: https://www.postgresql.org/ .(дата звернення: 18.05.2026).

18. Database for your application. URL: https://wearebrain.com/blog/how-to-choose-the-right-database-for-your-application/. (дата звернення: 27.04.2026).

19. My SQL. URL: https://www.mysql.com. (дата звернення: 27.04.2026).

20. Моринець С. NoSQL технології на прикладі MongoDB. *DOU*. URL: https://dou.ua/forums/topic/33453./ (дата звернення: 27.04.2026).

21. Redis Search. URL: https://redis.io/docs/latest/develop/ai/search-and-query/ (дата звернення: 27.04.2026).

22. CRM для бізнесу. Українська CPM-система для ecommerce та сфери послуг. Автоматизація бізнесу та спілкування з клієнтами KeyCRM URL: https://surli.cc/ehxwej. (дата звернення: 20.05.2026).

23. CRM (Customer Relationship Management) від Microsoft Dynamics 365. *Progresia*. URL: https://surli.cc/tfopqo. (дата звернення: 20.05.2026).

24. Salesforce CRM Platform: Customer 360 Overview. URL: https://www.salesforce.com/ (дата звернення: 11.05.2026).

25. Creatio – Low-code платформа для управління процесами та CRM. URL: https://www.creatio.com/uk (дата звернення: 12.05.2026).

26. Pipedrive CRM: Світовий лідер серед візуальних систем управління продажами. URL: https://www.pipedrive.com/ (дата звернення: 14.05.2026).

27. HubSpot. URL: https://www.hubspot.com (дата звернення: 27.04.2026).

28. Zoho CRM. URL: https://www.zoho.com (дата звернення: 27.04.2026).

29. Developmentframework. *3pillarglobal*.URL:https://www.3pillarglobal.com /insights/blog/key-decision-criteria-for-selecting-a-development-framework/(дата звернення: 27.04.2026).

30. Крилик Л., Шаргало Д. Розширення функціональних можливостей програмного модуля інформаційної технології управління відносинами з клієнтами. *Herald of Khmelnytskyi National University. Technical Sciences.* 2025. 347(1). С. 156-161.

31. Мірошниченко І. В., Алексєєва В. Д. Використання RFM-аналізу в сегментації клієнтів. *Економіка та держава.* 2022. № 1. С.114-122.

32. RFM segmentation. *Documentation.* URL: https://documentation.bloomreach.com/engagement/docs/rfm-segmentation. (дата звернення: 27.04.2026).

33. Shaji G. Customer segmentation using k means. *Pulse*. URL: https://www.linkedin.com/pulse/customer-segmentation-using-k-means-gopika-shaji-futnc/. (Lata звернення: 27.04.2026).

34. Boiko B., Protcyk I. Classification Model for Effective Employee Segmentation. *Моделювання, керування та інформаційні технології*. (7). 2025. С.60–61.

35. UML diagrams. URL: https://www.lucidchart.com/blog/types-of-UML-diagrams. (дата звернення: 27.04.2026).

36. Mebelki. Стильні та сучасні меблі. URL:https://mebelki.com.ua. (дата звернення: 12.05.2026).

37. Пасічник В. В., Пасічник О. В., Угрин Д. І. Веб-технології, підручник. Львів: «Магнолія 2006», 2025. 336 с.

38. Що таке JavaScript, та як функціонує JavaScript. URL: http://ruszura.in.ua/uncategorized/scho-take-javascript-yakfunktsionuyu javascript.html (дата звернення: 27.04.2026).

39. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript. (дата звернення: 27.04.2026).

40. Принципи тестування. URL: https://qalight.ua/baza-znaniy/printsipi-testuvannya . (дата звернення: 12.05.2026).

41. Зелінська О. В., Січко Т. В., Потапова Н. А., Якубич К. О. Методичні рекомендації до виконання, оформлення та захисту кваліфікаційної (бакалаврської) роботи здобувачами спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки». Вінниця: ДонНУ імені Василя Стуса, 2024. 27 с.

ДОДАТОК А
Лістинг коду

pages/Dashboard/Dashboard.jsx

```
import {useEffect,useState} from "react";
import axios from "axios";

function Dashboard(){

const [stats,setStats]=useState({

clients:0, orders:0, income:0

})

const [orders,setOrders]=useState([])

useEffect(()=>{

loadDashboard()

},[])

async function loadDashboard(){

try{

const statsResponse= await axios.get(
"http://localhost:5000/api/dashboard"
)

const ordersResponse= await axios.get( "http://localhost:5000/api/orders"
)
setStats(statsResponse.data)

setOrders( ordersResponse.data
)

}

catch(error){

console.log(error)

}

}

return(
```

```
<div>
<h1>
Панель продавця
</h1>
<div>
<div>
<h3>
Всього клієнтів
</h3>
<h1>
{stats.clients}
</h1>
</div>
<div>
<h3>
Всього замовлень
</h3>
<h1>
{stats.orders}
</h1>
</div>
<div>
<h3>
Загальний дохід
```

```
</h3>
<h1>
₴{stats.income}
</h1>
</div>
</div>
<h2>
Останні замовлення
</h2>
<table>
<thead>
<tr>
<th>
Номер
</th>
<th>
Клієнт
</th>
<th>
Сума
</th>
<th>
Статус
</th>
```

```
</tr>
</thead>
<tbody>
{
orders.map(order=>(
<tr key={order.id}>
<td>
#{order.id}
</td>
<td>
{order.client}
</td>
<td>
₴{order.total}
</td>
<td>
{order.status}
</td>
</tr>
))
}
</tbody>
</table>
</div>
```

```
)
}
export default Dashboard
<th>Категорія</th> <th>Ціна</th> <th>Кількість</th> <th>Статус</th> <th>Дії</th>


</tr>
</thead>
<tbody>
{
filteredProducts.map(product=>(
<tr key={product.id}>
<td>
{product.id}
</td>
<td>
{product.name}
</td>
<td>
{product.category}
</td>
<td>
${product.price}
</td>
```

```
<td>

{product.stock}

</td>

<td>

{product.status}

</td>

<td>

<button>
```

Components/AddProductModal/AddProductModal.jsx

```
import {useState} from "react";

import axios from "axios";

function AddProductModal({

setShowModal, loadProducts

}){

const[name,setName]= useState("")

const[price,setPrice]= useState("")

const[category,setCategory]= useState("") const[stock,setStock]=
useState("")

async function saveProduct(){

try{

await axios.post(

"http://localhost:5000/api/products",

{

name,
price, category, stock
```

```
}
)
loadProducts()
setShowModal(false)
}
catch(error){
console.log(error)
}
}
return(
<div>
<h2>
Усі статуси
</option>
<option>
Виконано
</option>
<option>
Очікується
</option>
<option>
Скасовано
</option>
</select>
<table>
```

```
<thead>
<tr>
<th>ID</th>
<th>Клієнт</th> <th>Дата</th> <th>Сума</th>
<th>Статус</th>
</tr>
</thead>
<tbody>
{ filteredOrders.map(order=>(
<tr key={order.id}>
<td>
#{order.id}
</td>
<td>
{order.client}
</td>
<td>
{order.date}
</td>
<td>
₴{order.total}
</td>
<td>
{order.status}
```

```
</td>

</tr>

))

}

</tbody>

</table>

</div>

)
}

export default Orders
```

Pages/Clients/Clients.jsx

```
import {useEffect,useState} from "react";

import axios from "axios";

import AnalyticsModal from
"../../components/AnalyticsModal/AnalyticsModal";

function Clients(){

const[clients,setClients]= useState([])

const[selectedClient, setSelectedClient]=
useState(null)

useEffect(()=>{

loadClients()

},[])

async function loadClients(){

try{

const response=
```

```
await axios.get(

"http://localhost:5000/api/clients"

)

setClients(

response.data

)
}

catch(error){
console.log(error)

}

}

return(

<div>

<h1>

Аналітика

</button>

</td>

</tr>

))

}

</tbody>

</table>

{

selectedClient &&

<AnalyticsModal
```

```
client={selectedClient}

setSelectedClient={

setSelectedClient

}

/>

}

</div>

)
}

export default Clients
```

Components/AnalyticsModal/AnalyticsModal.jsx

```
function AnalyticsModal({

client, setSelectedClient

}){

return(

<div>

<h1>

Аналітика клієнта

</h1>

<hr/>

<h3>

Загальна інформація

</h3>

<p>

Ім'я:
```

```
{client.name}

</p>

<p>

Тип:

{client.type}

</p>
<p>

Усього замовлень:

{client.orders}

</p>

<p>

Витрачено:

${client.spent}

</p>

<p>

Середній чек:

${client.average}

</p>

<hr/>

<h3>

Поведінкові метрики

</h3>

<p>

Переглянуто товарів:

12
```

Додано в кошик:

5

Покупок:

3

Конверсія:

25%

<hr/>

<h3>

Найбільший інтерес

</h3>

Категорія:

Вітальня

Товар:

Диван

```
<p>

Інтерес:

Високий
</p>

<hr/>

<button

onClick={()=>{

setSelectedClient(null)

}}

>

Закрити

</button>

</div>

)

}

export default AnalyticsModal
Controllers/clientsController.js

const pool= require( "../db"
)

function determineClientType(

orders

){
  if(

orders===0

){

return
```

```
"Потенційний"
}   if(
orders<=4
){
return
"Постійний"
}
return "VIP"
}
exports.getClients=
async(
req, res
)=>{
try{
const clients=
await pool.query(
`
SELECT
users.id,
users.name,
COUNT(
orders.id
)
as orders,
```

```
COALESCE(
SUM(
orders.total
),
0
)
as spent
FROM users
LEFT JOIN orders
ON
users.id=
orders.user_id
```