

УДК 004.415:005

РОЗРОБКА ВЕБОРІЄНТОВАНОЇ CRM-СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ КЛІЄНТСЬКИМИ ДАНИМИ

Пугачьов І.М.

e-mail: illia.puhachov1@nure.ua

Науковий керівник – старший викладач Губін В.О.

Харківський національний університет радіоелектроніки, каф. III
м. Харків, Україна

This work presents the design and implementation of a web-oriented CRM system for automating lead management in marketing. The system provides flexible lead routing, campaign management, role-based access control, and automated postback tracking. The architecture follows a client-server model with Next.js on the frontend and Django on the backend, PostgreSQL and Celery for asynchronous processing. API documentation makes all available methods and data fields, simplifying third-party integration.

Ефективне управління клієнтськими відносинами є одним із ключових факторів успіху сучасного бізнесу. Зі зростанням кількості контактів, угод і партнерів компанії стикаються з проблемою розрізненості даних: інформація зберігається в таблицях, месенджерах і електронній пошті одночасно, а її обробка потребує значних часових витрат.

CRM (Customer Relationship Management) системи дозволяють централізувати клієнтські дані, автоматизувати рутинні операції та отримати прозору аналітику по воронці продажів.

Метою роботи є проєктування та розробка власної CRM системи, яка була б адаптована до партнерського маркетингу й управління клієнтськими даними, з гнучкою архітектурою та можливістю подальшого розширення за рахунок додавання застосунків Django.

В роботі був проведений детальний аналіз ключових вимог до системи. Основними критеріями стали: висока пропускна здатність при прийомі даних, гнучке налаштування маршрутизації, підтримка партнерських інтеграцій, а також можливість брендування кампаній під конкретних клієнтів.

Окремою вимогою стала підтримка власних доменів для кожної, що дозволяє розгортати захищені вебсторінки без ручного адміністрування сертифікатів. Також було визначено необхідність відкритої API-документації, яка описує всі доступні методи та поля – це суттєво знижує поріг входу для сторонніх розробників та партнерів, що інтегруються з системою.

В роботі досліджується архітектура розробленої CRM-системи.

Розроблена система побудована за клієнт-серверною архітектурою з чітким розділенням відповідальності між рівнями. Фронтенд-частина реалізована як односторінковий застосунок (SPA) на базі Next.js

і React, що забезпечує швидке завантаження інтерфейсу без зайвих перезавантажень сторінки.

Серверна частина побудована на Django з використанням Django REST Framework для реалізації API-шару. Для виконання фонових завдань (відправки постбеків і обробки конверсій) застосовується Celery у зв'язці з Redis як брокером черг. Підтримка ASGI і WebSocket через Daphne забезпечує можливість отримання оновлень у реальному часі.

Для зберігання даних використовується PostgreSQL – реляційна СУБД, що добре підходить для складних запитів з фільтрацією та агрегацією. Nginx виконує роль зворотного проксі та обслуговує статику через WhiteNoise. Вся інфраструктура запакована у Docker-контейнери і керується через Docker Compose, що гарантує ідентичність середовищ розробки та продакшн.

Система включає декілька ключових функціональних модулів, кожен з яких відповідає за окремий аспект роботи з клієнтськими даними.

Розглянемо їх більш детально.

Модуль управління лідами дозволяє відстежувати повний шлях потенційного клієнта від першого контакту до закриття угоди. Для кожного ліда можна встановити статус, додати теги, налаштувати кастомні поля та зберегти додаткові дані в довільній структурі.

Модуль кампаній та потоків надав можливість створювати кампанії(джерела даних) з розділення на дочірні потоки, це дозволяє робити розподілення та додавання різних партнерів безпосередньо через CRM. Кожен потік приймає ліди з своїми власними налаштуваннями, що дозволяє сегментувати потоки вхідних заявок і застосовувати різну логіку обробки.

Модуль маршрутизації (роутингу) є одним із ключових елементів системи. Він дозволяє сортувати вхідні заявки за будь-якими полями (гео, джерелом, мовою, значеннями кастомних параметрів тощо) і миттєво перемикає потоки між партнерами без зупинки прийому лідів. Завдяки функціонуванню цього модуля система в штатному режимі обробляє до 5000 заявок на годину.

Модуль створення інтеграцій надає можливість конструювати стек запитів з підтримкою різних методів, цей модуль використовує JMESPath з шаблонізатором даних. Як показала практика цей метод є дуже універсальним та дуже зручним.

Модуль партнерів та інтеграцій забезпечує маршрутизацію даних між партнерами з урахуванням денних і місячних лімітів, мапінг статусів між внутрішньою логікою та очікуваними значеннями на стороні партнера.

Модуль постбеків та трекінгу автоматизує відправку сповіщень про конверсії на зовнішні системи. Журнал постбеків фіксує всі спроби відправки, їхній результат і час виконання, що спрощує діагностику збоїв.

Модуль ролей і прав доступу реалізує ієрархічну модель керування користувачами. Адміністратор визначає ролі з точним набором дозволів, розподіляє команди та обмежує видимість даних на рівні окремих модулів – від перегляду статистики до управління партнерами.

Модуль аналітики та звітності надає гнучкі дашборди з фільтрами за будь-якими параметрами.

Дані можна експортувати у зовнішні інструменти, а налаштовані метрики дозволяють кожній команді бачити саме ті показники, що є важливими для її завдань.

API-документація формується автоматично на основі коду через drf-spectacular і містить повний опис усіх методів, параметрів запитів та полів відповідей. Це суттєво спрощує інтеграцію з партнерськими системами та розробку власних клієнтів.

Впровадження розробленої CRM-системи дозволило досягти відчутних практичних результатів. Час обробки вхідного ліда скоротився завдяки автоматичній маршрутизації та фоновій відправці постбеків. Прозора воронка продажів з детальною фільтрацією підвищила контроль над конверсією на кожному етапі.

Модульна архітектура системи забезпечила просту масштабованість: додавання нового партнера або нового поля до картки ліда не вимагає змін у базовому коді. Навантажувальне тестування підтвердило стабільність платформи навіть у пікових умовах.

Якщо розглядати платформу з точки зору використання технологій штучного інтелекту (ШІ), то вона є дуже потужною основою для побудови системи, яка буде відповідати сучасним вимогам: аналізу даних з використанням LLM; можливості завдяки ШІ приймати обґрунтовані рішення щодо якості даних, опираючись на аналіз та виділення патернів, які можуть характеризувати фрод.

Також, в подальшому, пропонується більш детально дослідити питання взаємодії ШІ з реальними клієнтами у контексті обробки їх запитів.

Список використаних джерел:

1. Django Documentation: official documentation. URL: <https://docs.djangoproject.com/en/6.0/> (дата звернення 05.03.2026).
2. Next.js Documentation: official documentation. URL: <https://nextjs.org/docs> (дата звернення 05.03.2026).
3. ShadCN Documentation: official documentation URL: <https://ui.shadcn.com/docs> (дата звернення 05.03.2026).
4. Redis Documentation: official documentation. URL: <https://redis.io/docs> (дата звернення 05.03).