

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) Інформаційних технологій та електроніки

Кафедра Інформаційних технологій та програмування
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр
(бакалавр, магістр)

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

на тему Система керування взаємодії з клієнтами для малого та середнього бізнесу

Виконав: студент групи ІПЗ-20д

_____ (підпис)

Є. Д. Малишев
(ініціали і прізвище)

Керівник

_____ (підпис)

Д. М. Марченко
(ініціали і прізвище)

Завідувач кафедри

_____ (підпис)

О. І. Захожай
(ініціали і прізвище)

Рецензент О. І. Захожай _____

Київ – 2024

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) Інформаційних технологій та електроніки.

Кафедра Інформаційних технологій та програмування

(повна назва кафедри)

Освітній ступінь бакалавр

(бакалавр, магістр)

спеціальність 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Захожай О. І.

“ _____ ” _____ 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ

Малишев Єгор Дмитрович

(прізвище, ім'я, по-батькові)

1. Тема роботи Система керування взаємодії з клієнтами для малого та середнього бізнесу

Керівник роботи Марченко Дмитро Миколайович, професор, д.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом університету від “ 06 ” травня 2024 року № 171/15.15-С

2. Строк подання студентом роботи 08.06.2024

3. Вихідні дані до роботи _____

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

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників) _____

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання 30.03.2024

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання кваліфікаційної випускної роботи | Строк виконання етапів | Примітка |
|-------|---|------------------------|----------|
| 1 | Дослідження предметної області | 01.04.24-08.04.24 | виконано |
| 2 | Пошук та аналіз існуючих рішень | 09.04.24-16.04.24 | виконано |
| 3 | Аналіз вимог до системи | 17.04.24-24.04.24 | виконано |
| 4 | Проектування системи | 25.04.24-04.05.24 | виконано |
| 5 | Розробка системи | 05.05.24-17.05.24 | виконано |
| 6 | Тестування системи | 18.05.24-25.05.24 | виконано |
| 7 | Оформлення пояснювальної записки | 26.05.24-05.06.24 | виконано |
| 8 | Підготовка та подання дипломної роботи | 06.06.24-08.06.24 | виконано |

Студент

(підпис)

Є. Д. Малишев

(ініціали і прізвище)

Керівник роботи

(підпис)

Д. М. Марченко

(ініціали і прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ

дипломної роботи студента гр. ІПЗ-20д Малишев Є. Д.

Науковий керівник

Професор, д.т.н.

Марченко Д.М.

Оцінка наукового керівника: _____

Рецензент Захожай О. І., д.т.н., в.о.зав.каф.ІТП СНУ ім. В.Даля
ПБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

Професор кафедри ІТП

д.т.н.

підпис

Меняйленко О.С.

ЗМІСТ

| | |
|---|----|
| ВСТУП | 7 |
| РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ | 9 |
| 1.1. Поняття CRM-системи та її роль у керуванні взаємодією з клієнтами .. | 9 |
| 1.2. Функціональні можливості CRM-систем..... | 9 |
| 1.3. Класифікація CRM-систем | 10 |
| 1.4. Огляд існуючих CRM систем для малого та середнього бізнесу | 10 |
| 1.4.1 Salesforce | 10 |
| 1.4.2 HubSpot CRM..... | 12 |
| 1.4.3 SalesDrive | 14 |
| 1.4.4 KeyCRM | 16 |
| РОЗДІЛ 2. ФУНКЦІОНАЛЬНІ ВИМОГИ..... | 18 |
| 2.1. Основні функціональні вимоги | 18 |
| 2.2. Нефункціональні вимоги | 19 |
| 2.3. Інтерфейс та юзабіліті | 20 |
| 2.4. Інтеграції та навантаження | 21 |
| 2.5. Безпека та захист даних | 22 |
| РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ | 23 |
| 3.1. Вибір мови програмування та фреймворку | 23 |
| 3.1.1 Python..... | 23 |
| 3.2 Клієнтська частина | 24 |
| 3.2.1 HTML..... | 24 |
| 3.2.2 CSS..... | 24 |
| 3.3 Серверна частина | 25 |
| 3.3.1 Django | 25 |
| 3.4 База даних..... | 26 |
| 3.4.1 SQLite | 26 |
| 3.5 Проєктування CRM-системи | 27 |
| 3.2.1 Архітектура системи..... | 27 |
| 3.6 Основні компоненти системи..... | 28 |
| 3.7 Розробка модулів CRM-системи | 30 |
| 3.7.1 Модуль керування користувачами | 31 |
| 3.7.2 Модуль керування контактами | 32 |
| 3.7.3 Модуль керування угодами..... | 34 |
| 3.7.4 Модуль автоматизації маркетингу | 35 |
| 3.7.5 Модуль обслуговування клієнтів..... | 36 |

| | |
|---|----|
| 3.7.6 Модуль аналітики та звітності | 37 |
| РОЗДІЛ 4. ТЕСТУВАННЯ СИСТЕМИ | 38 |
| ВИСНОВОК..... | 42 |
| СПИСОК ЛІТЕРАТУРИ..... | 44 |
| ДОДАТОК А..... | 46 |

ВСТУП

У сучасному бізнес-середовищі, де конкуренція стає все більш інтенсивною, ефективне керування взаємодією з клієнтами є критичним фактором успіху для малих та середніх підприємств. CRM-системи (системи керування взаємодією з клієнтами) відіграють ключову роль у забезпеченні цього процесу, допомагаючи організаціям не лише залучати нових клієнтів, а й утримувати наявних, підвищуючи рівень їх задоволеності та лояльності.

Актуальність даної теми обумовлена необхідністю підвищення ефективності бізнес-процесів малих та середніх підприємств за рахунок впровадження сучасних технологій керування взаємодією з клієнтами. З розвитком інформаційних технологій все більше підприємств визнають важливість CRM-систем для оптимізації своєї роботи та поліпшення відносин з клієнтами. Використання мови програмування Python та фреймворку Django дозволить створити професійний, якісний та актуальний інструмент, який здатний задовольнити потреби малого та середнього бізнесів.

Метою кваліфікаційної роботи є розробка ефективної CRM-системи для малого та середнього бізнесів, використовуючи мову програмування Python та фреймворку Django.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Провести огляд предметної області, визначивши роль та функціональні можливості CRM-систем.
2. Аналіз існуючих CRM-рішень для малого та середнього бізнесів, визначення їх переваг та недоліків конкурентних систем.
3. Визначити функціональні та нефункціональні вимоги до розроблюваної CRM-системи.

4. Розробити архітектуру та модулі CRM-системи на основі Python та Django.
5. Провести апробацію розробленої CRM-системи.

Наукова новизна даної роботи полягає в застосуванні сучасних технологій Python та Django для розробки CRM-системи, яка орієнтована на потреби малого та середнього бізнесу. В процесі дослідження буде створено унікальну архітектуру системи, що забезпечить високий рівень інтеграції, зручності та безпеки. Також буде проведено детальний аналіз існуючих рішень, що буде корисним для з'ясування вимог користувачів та методів реалізації системи. Таким чином, ця розробка сприятиме підвищенню ефективності управління взаємодією з клієнтами в малому та середньому бізнесі, надаючи їм інструмент, який відповідає сучасним вимогам ринку.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття CRM-системи та її роль у керуванні взаємодією з клієнтами

CRM-система — це аббревіатура з англ. Customer Relationship Management, вона є програмним забезпеченням, призначеним для керування взаємодією з клієнтами, організації бізнес-процесів, підвищення ефективності продажів і маркетингу[1]. CRM-система дозволяє збирати, аналізувати та зберігати інформацію про клієнтів, їхні потреби та поведінку, це сприяє кращому розумінню клієнтської бази та підвищенню рівня обслуговування.

Основними завданнями CRM-систем є:

- Автоматизація процесів продажу, маркетингу та обслуговування клієнтів.
- Збереження історії взаємодії з клієнтами.
- Управління контактами та угодами.
- Підвищення рівня задоволеності та лояльності клієнтів.
- Оптимізація внутрішніх бізнес-процесів

1.2. Функціональні можливості CRM-систем

CRM-системи, залежно від потреб, пропонують широкий спектр функціональних можливостей[2], серед яких:

- **Управління контактами** (створення та підтримка детальної бази даних клієнтів з інформацією про кожного клієнта).
- **Управління продажами** (відстеження всіх етапів продажу, управління потенційними клієнтами та прогнозування продажів).
- **Автоматизація маркетингу** (проведення маркетингових кампаній, аналіз ефективності та сегментація клієнтів).
- **Обслуговування клієнтів** (забезпечення швидкого та ефективного обслуговування клієнтів через різні канали зв'язку).

- **Аналітика та звітність** (створення звітів та аналізу даних для прийняття обґрунтованих рішень).

1.3. Класифікація CRM-систем

CRM-системи можна класифікувати за різними критеріями, в залежності від потреб бізнесу, та його специфікації[3]:

1. За типом розгортання (хмарні (SaaS) та локальні (On-Premise)).
2. За функціональними можливостями (операційні, аналітичні та колаборативні).
 - Операційні CRM-системи (зосереджуються на автоматизації бізнес-процесів, пов'язаних з продажами, маркетингом та обслуговуванням клієнтів).
 - Аналітичні CRM-системи (орієнтовані на аналіз даних про клієнтів та ринок для прийняття стратегічних рішень).
 - Колаборативні CRM-системи (забезпечують взаємодію між різними підрозділами компанії та клієнтами через різні комунікаційні канали).

1.4. Огляд існуючих CRM систем для малого та середнього бізнесу

На сьогоднішній день на ринку представлено безліч CRM-систем, що орієнтовані на малий та середній бізнес, серед яких:

1.4.1 Salesforce

Salesforce[4] американська хмарна CRM-система, що пропонує широкий спектр функціональностей для управління продажами, маркетингом та обслуговуванням клієнтів.

Основні функції Salesforce:

1. Управління контактами та лідами

- Збір, зберігання та організація даних про клієнтів, відстеження взаємодії з клієнтами, включаючи історію контактів та запити.

2. Управління продажами

- відстеження угод на всіх етапах продажу
- управління комерційними пропозиціями та контрактами
- інструменти для прогнозування продажів та аналізу ефективності.

3. Автоматизація маркетингу

- створення та управління маркетинговими кампаніями
- сегментація клієнтів для таргетованих розсилок
- відстеження результатів кампаній та аналіз ефективності.

4. Аналітика та звітність

- гнучкі інструменти для створення звітів та дашбордів
- аналіз ключових показників бізнесу.
- візуалізація даних для прийняття обґрунтованих рішень.

5. Додаткові сервіси

- підтримка інтеграцій з різними сервісами та платформами.

6. Мобільний доступ

- доступ до CRM з мобільних пристроїв через додатки для iOS та Android.

7. Автоматизація процесів

- інструменти для автоматизації задач.

Переваги та недоліки Salesforce:

- **Переваги:** великий функціонал, масштабованість, інтеграція з великою кількістю інших систем, такі як аналітика та телефонія.
- **Недоліки:** висока вартість, складність в налаштуванні та використанні для невеликих компаній, створена для західного ринку.

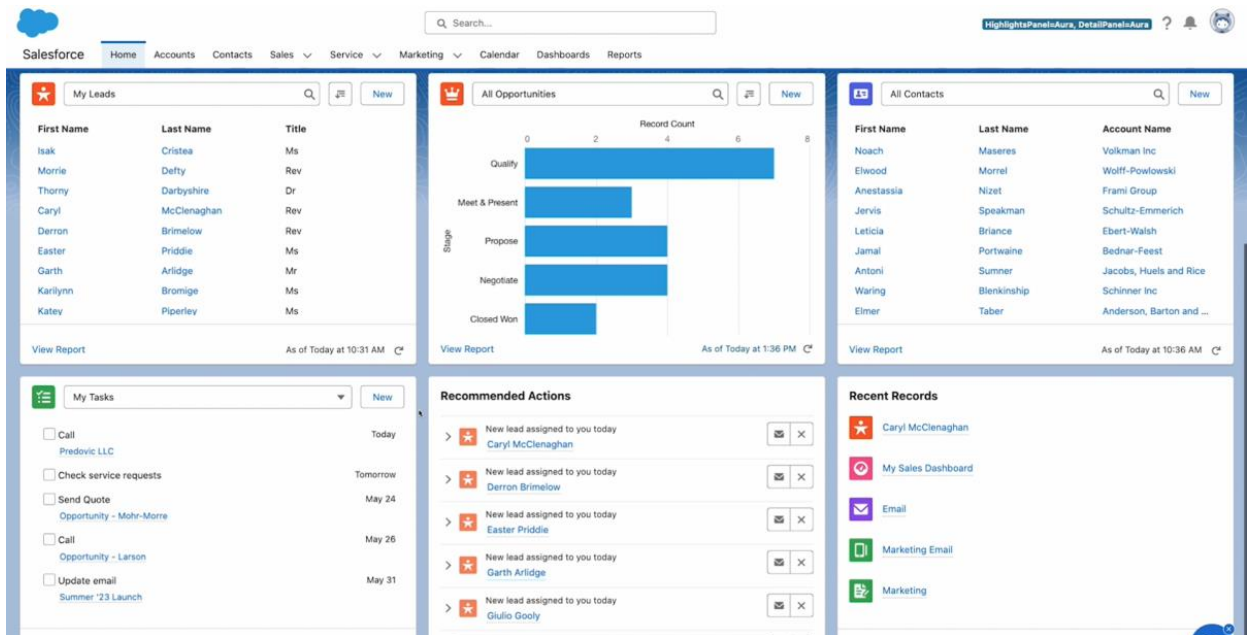


Рис. 1.1 — Salesforce

1.4.2 HubSpot CRM

HubSpot CRM[5] є безкоштовною CRM-системою, що забезпечує базові можливості для управління контактами, угодами та маркетинговими кампаніями.

Основні функції HubSpot CRM:

1. Управління контактами та лідами

- збір, зберігання та організація контактної інформації.
- автоматичне відстеження взаємодії з клієнтами, включаючи електронну пошту та телефонні дзвінки.

2. Управління продажами

- відстеження угод та їхніх етапів у воронці продажів.
- інструменти для планування задач та зустрічей.
- автоматичне створення звітів про ефективність продажів.

3. Автоматизація маркетингу

- інструменти для створення та управління маркетинговими кампаніями.
- сегментація контактів для таргетованих розсилок.

- відстеження ефективності продажів та аналіз показників.

4. Аналітика та звітність

- інтерактивні дашборди та звіти для аналізу діяльності.
- відстеження ключових показників бізнесу та ефективності маркетингових та продажних активностей.
- інструменти для візуалізації даних.

5. Додаткові сервіси

- підтримка інтеграцій з іншими інструментами маркетингу, та продажів.
- інтеграції з сторонніми сервісами, такими як електронна пошта, соціальні мережі, та інші бізнес-інструменти.
- використання API для створення власних інтеграцій.

6. Мобільний доступ

- доступ до CRM через мобільні додатки для iOS та Android.
- можливість працювати з даними в будь-який час та з будь-якого місця.

7. Інструменти для продажів

- вбудовані інструменти для створення комерційних пропозицій.
- автоматизація процесу відправки та відстеження електронних листів.
- інструменти для планування та проведення зустрічей.

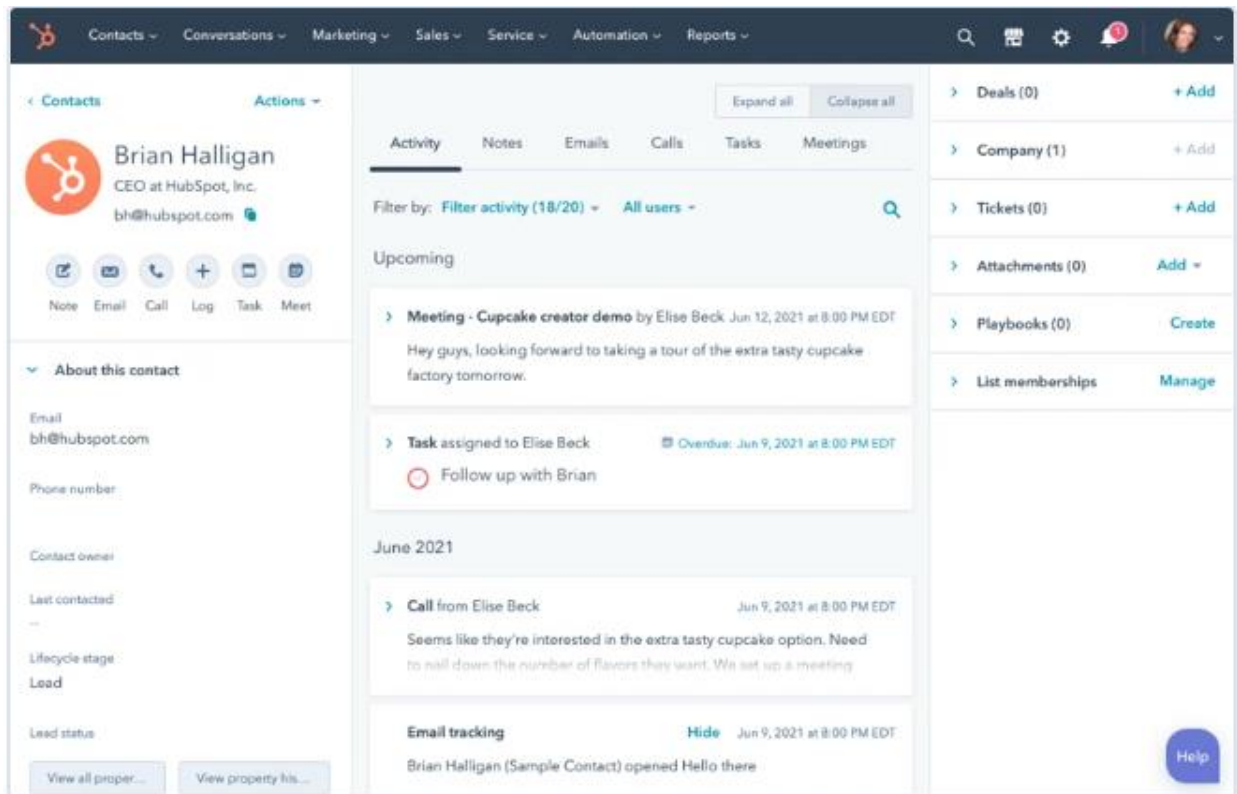
8. Підтримка та обслуговування клієнтів

- управління запитамі клієнтів через систему тикетів
- база знань для самообслуговування клієнтів.
- чат-боти та інші інструменти для підтримки в реальному часі.

Переваги та недоліки HubSpot CRM:

Переваги: безкоштовна основна версія, простота у використанні, інтеграція з маркетинговими інструментами HubSpot (такі як аналітика, телефонія та воронки продажів).

Недоліки: обмежені можливості безкоштовної версії, висока вартість додаткових функцій, створена для західного та європейського ринків.



1.2 — HubSpot CRM

1.4.3 SalesDrive

SalesDrive[6] — це CRM-система, створена для українського ринку, орієнтована на автоматизацію процесів продажу і маркетингу. Вона вміщує в собі широкий спектр функціональних можливостей для малого та середнього бізнесу, забезпечуючи ефективне управління взаємодією з клієнтами.

Основні функції SalesDrive:

1. Управління контактами

- зручний інтерфейс для ведення бази даних клієнтів, включаючи інформацію про контакти, історію взаємодії та важливі дати.

2. Управління продажами

- автоматизація процесу продажів, від створення лідів до завершення угод. Система підтримує створення комерційних пропозицій, виставлення рахунків та контроль за виконанням угод.

3. Аналітика та звітність

- можливість створення звітів та аналізу даних, що дозволяє відслідковувати ефективність роботи відділу продажів та маркетингу.

4. Додаткові системи

- підтримка інтеграцій з допоміжними сервісами, такими як електронна пошта, телефонія, месенджери та соціальні мережі.

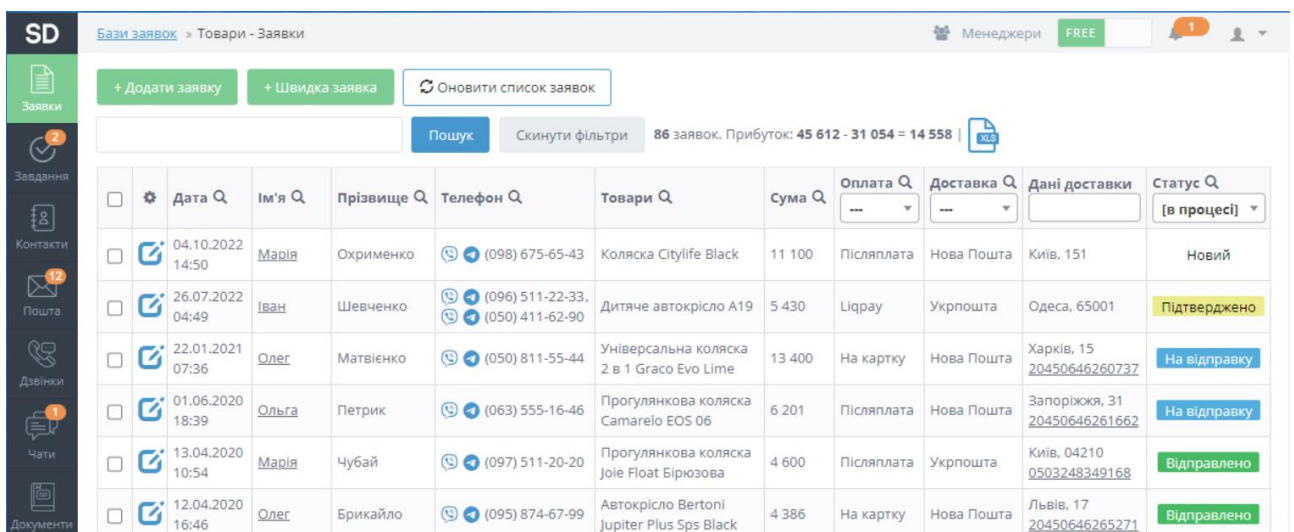
5. Автоматизація маркетингу

- інструменти для створення та управління маркетинговими кампаніями, розсилок та сегментації клієнтів.

Переваги та недоліки SalesDrive:

Переваги: легке підключення з інших сервісів та систем, індивідуальне налаштування під потреби малого та середнього бізнесу, докладні аналітичні інструменти для оцінки ефективності роботи.

Недоліки: висока вартість підключення, необхідне додаткове навчання персоналу для взаємодії з системою.



| <input type="checkbox"/> | Дата | Ім'я | Прізвище | Телефон | Товари | Сума | Оплата | Доставка | Дані доставки | Статус |
|--------------------------|---------------------|-------|-----------|-------------------------------------|--|--------|------------|------------|---------------------------------|--------------|
| <input type="checkbox"/> | 04.10.2022 14:50 | Марія | Охрименко | (098) 675-65-43 | Коляска Citylife Black | 11 100 | Післяплата | Нова Пошта | Київ, 151 | Новий |
| <input type="checkbox"/> | 26.07.2022 04:49 | Іван | Шевченко | (096) 511-22-33, (050) 411-62-90 | Дитяче автокрісло A19 | 5 430 | Літрау | Укрпошта | Одеса, 65001 | Підтверджено |
| <input type="checkbox"/> | 22.01.2021 07:36 | Олег | Матвієнко | (050) 811-55-44 | Універсальна коляска 2 в 1 Graco Evo Lime | 13 400 | На картку | Нова Пошта | Харків, 15 20450646260737 | На відправку |
| <input type="checkbox"/> | 01.06.2020 18:39 | Ольга | Петрик | (063) 555-16-46 | Прогуляньова коляска Camarelo EOS 06 | 6 201 | Післяплата | Нова Пошта | Запоріжжя, 31 20450646261662 | На відправку |
| <input type="checkbox"/> | 13.04.2020 10:54 | Марія | Чубай | (097) 511-20-20 | Прогуляньова коляска Joie Float Бірюзова | 4 600 | Післяплата | Укрпошта | Київ, 04210 0503248349168 | Відправлено |
| <input type="checkbox"/> | 12.04.2020 16:46 | Олег | Брикайло | (095) 874-67-99 | Автокрісло Bertoni Jupiter Plus Sps Black | 4 386 | На картку | Нова Пошта | Львів, 17 20450646265271 | Відправлено |

Рис. 1.3 — SalesDrive

1.4.4 KeyCRM

KeyCRM[7] — це українська CRM-система, спеціально розроблена для малого та середнього бізнесу України, що забезпечує комплексний підхід до управління і взаємодії з клієнтами.

Основні функції KeyCRM:

1. Управління контактами та лідами

- збір та зберігання інформації про клієнтів та потенційних клієнтів, автоматичне оновлення даних.

2. Управління продажами

- автоматизація процесу продажів, включаючи створення лідами,
- відстеження стадій угод та планування заходів.

3. Маркетингові кампанії

- планування, запуск та аналіз маркетингових кампаній, що дозволяє оптимізувати маркетингову стратегію.

4. Аналітика та звітність

- інструменти для створення детальних звітів та аналітики, що допомагають приймати обґрунтовані рішення.

5. Інтеграції

- підтримка інтеграцій з різними сервісами та платформами, такими як електронна пошта, соціальні мережі та інші бізнес-інструменти.

6. Підтримка мобільних пристроїв

- доступ до системи з мобільних пристроїв, що забезпечує мобільність та гнучкість роботи.

Переваги та недоліки KeyCRM:

Переваги: інтуїтивний та зручний інтерфейс, широкі можливості інтеграції з іншими системами, підтримка мобільних пристроїв для доступу в будь-який час та з будь-якого місця.

Недоліки: можливість недостатньої адаптивності для дуже специфічних потреб бізнесу, вартість може бути високою для стартапів та дуже малих компаній.

| Замовлення | Покупець | Джерело | Фіскалізація | Час створення | Дата доставки | Артикул товару | Сума за товари | Загальна вартість | Статус |
|------------|-------------------|---------|------------------------|------------------|---------------|----------------|----------------|-------------------|--------------------------|
| 2831 | Дмитро Малирчук | Rozetka | Не фіскалізов. | Вчора 23:56 | + Додати | n-02s | 235,00 грн | 235,00 грн | ВІДПРАВЛЕНО НОВОЮ ПОШТОЮ |
| 2830 | Людмила Федосенко | Prom.ua | Не фіскалізов. | Вчора 15:48 | + Додати | n-16 | 200,00 грн | 200,00 грн | ВІДМІНЕНО |
| 2829 | Евгенія Коршукова | Rozetka | Фіскалізація відмінена | 30.12.2021 20:31 | + Додати | — | 325,00 грн | 325,00 грн | ВИКОНАНО |
| 2828 | Cody Fisher | Amazon | Фіскалізований | 29.12.2021 19:07 | + Додати | — | \$47,50 | \$47,50 | НОВИЙ |
| 2827 | Igor Efanov | Prom.ua | Не фіскалізов. | 28.12.2021 13:45 | + Додати | p-40 | 450,00 грн | 450,00 грн | НЕДОСТАТНЬО ІНФОРМАЦІЯ |
| 2826 | Samuel Jackson | HiStore | — | 25.12.2021 04:31 | + Додати | N05 | \$14,00 | \$14,00 | ВИКОНАНО |

Рис. 1.4 — KeyCRM

Розробка власної CRM-системи на основі Python та Django дозволить врахувати специфічні потреби малого та середнього бізнесу, створити гнучке, професійне та економічно ефективне рішення.

РОЗДІЛ 2. ФУНКЦІОНАЛЬНІ ВИМОГИ

2.1. Основні функціональні вимоги

Функціональні вимоги – це спеціалізовані дії або функції, які система повинна виконувати для зручності користування та виконання потреб користувачів[8]. Вони визначають, що саме повинна робити система, описуючи її поведінку та взаємодію з користувачем у різних сценаріях. Функціональні вимоги включають в себе детальні описи завдань, процесів, бізнес-правил і умов, що стосуються роботи системи.

При розробці CRM-системи для малого та середнього бізнесу необхідно враховувати основні функціональні вимоги, що забезпечують ефективне управління взаємодією з клієнтами, а саме:

1. Управління контактами

- створення та ведення бази даних клієнтів з інформацією про кожного клієнта, включаючи контактні дані, історію взаємодії та уподобання.

2. Управління угодами

- відстеження процесу продажів від початкового контакту до завершення угоди, включаючи управління лідами та можливостями.

3. Автоматизація маркетингу

- проведення маркетингових кампаній, сегментація клієнтів, управління електронною поштою та відстеження результатів кампаній.

4. Обслуговування клієнтів

- надання засобів для обробки запитів клієнтів, управління, підтримка та відстеження часу вирішення проблем.

5. Аналітика та звітність

- створення та управління звітами для аналізу ефективності продажів, маркетингових кампаній та обслуговування клієнтів.

6. Календар та завдання

- управління завданнями та подіями, інтеграція з календарями для відстеження важливих подій та дедлайнів.

Функціональні вимоги є основою для розробки та впровадження системи[8], оскільки вони визначають, які задачі повинна виконувати система для задоволення потреб користувачів і досягнення бізнес-цілей. Чітке визначення та документування функціональних вимог допомагає забезпечити успішне впровадження системи, зменшуючи ризики помилок.

2.2. Нефункціональні вимоги

Нефункціональні вимоги – це характеристики та атрибути системи, які визначають як система повинна виконувати свої функції і включають такі аспекти, як продуктивність, надійність, безпека, масштабованість та інші[8].

Нефункціональні вимоги забезпечують загальну якість та користувацький досвід при взаємодії з системою.

Крім основних функціональних вимог, CRM-система повинна відповідати ряду нефункціональних вимог, які забезпечують її надійність, безпеку та зручність використання[9]:

- **Продуктивність** (система повинна забезпечувати швидку обробку запитів та високу швидкість роботи навіть при великій кількості користувачів).
- **Навантаження** (можливість масштабування системи у разі зростання кількості користувачів або обсягів даних).
- **Надійність** (висока доступність та стійкість до збоїв, регулярне резервне копіювання даних).
- **Безпека** (захист даних від несанкціонованого доступу, шифрування даних, аутентифікація користувачів).

- **Зручність використання** (інтуїтивно зрозумілий інтерфейс, що забезпечує зручне та швидке виконання основних операцій).
- **Додаткові сервіси** (підтримка інтеграції з іншими системами та сервісами, що використовуються в бізнесі).

Нефункціональні вимоги є критичними для успішного впровадження та експлуатації системи, оскільки вони визначають її якісні характеристики, які безпосередньо впливають на користувацький досвід і загальну ефективність системи[9]. Забезпечення відповідності нефункціональним вимогам допомагає гарантувати, що система буде надійною, продуктивною, безпечною та зручною у використанні, відповідаючи очікуванням користувачів та бізнес-цілям організації.

2.3. Інтерфейс та юзабіліті

При розробці інтерфейсу CRM-системи потрібно враховувати багато факторів, щоб забезпечити зручність використання та ефективність роботи для всіх категорій користувачів[10]. Якісний та зручний інтерфейс є одним із ключових факторів, що впливають на успішність впровадження та подальше використання CRM-системи. Він сприяє підвищенню продуктивності працівників та зниженню кількості помилок.

Розробка інтерфейсу CRM-системи повинна враховувати наступні аспекти:

1. Простота та інтуїтивність

- інтерфейс має бути зрозумілим для користувачів з різним рівнем технічної підготовки, з мінімальною кількістю кроків для виконання основних операцій.

2. Адаптивний дизайн

- система повинна бути доступною з різних пристроїв (ПК, планшети, смартфони), забезпечуючи однакову зручність використання.

3. Налаштування під потреби користувача

- можливість налаштування інтерфейсу відповідно до індивідуальних потреб користувачів, створення персоналізованих панелей управління та звітів.

Таким чином, зручний, зрозумілий інтерфейс та юзабіліті є ключовими аспектами, які визначають успішність використання CRM-системи та її вплив на бізнес-процеси організації[10].

2.4. Інтеграції та навантаження

CRM-система повинна забезпечувати можливість інтеграції з іншими бізнес-системами та сервісами[11], це дозволить оптимізувати бізнес-процеси, підвищити ефективність роботи та забезпечити зручність використання. Крім того, важливим аспектом є можливість витримувати велике навантаження, що забезпечує можливість системи рости разом з бізнесом, адаптуючись до змін в обсягах даних та кількості користувачів.

CRM-система повинна забезпечувати можливість інтеграції з іншими бізнес-системами та сервісами:

1. Інтеграція з електронною поштою

- автоматичне збереження історії листування з клієнтами, можливість надсилання листів безпосередньо з системи.

2. Інтеграція з телефонією

- можливість здійснення та запису телефонних дзвінків, автоматичне збереження історії дзвінків.

3. Інтеграція з бухгалтерськими системами

- автоматичне створення рахунків-фактур, управління платежами та фінансовими документами.

4. Інтеграція з маркетинговими платформами

- управління маркетинговими кампаніями, відстеження результатів та аналітика.

Забезпечення можливості інтеграції та робота з великим обсягом даних CRM-системи є ключовими аспектами її ефективного використання у малому та середньому бізнесі[11]. Вони дозволяють оптимізувати бізнес-процеси, покращувати комунікацію та управління клієнтськими взаємодіями, забезпечуючи тим самим зростання та розвиток бізнесу.

2.5. Безпека та захист даних

Безпека та захист даних є критично важливими аспектами розробки CRM-системи[12]. В сучасних умовах зростання цифрових продуктів та цифрової трансформації, коли обсяги даних та кількість користувачів зростають, кіберзагрози стають все більш небезпечними та складними, забезпечення конфіденційності, цілісності та доступності інформації є першочерговим завданням. Надійна система безпеки повинна захищати не тільки дані компанії, але й особисту інформацію клієнтів, забезпечуючи довіру та відповідність нормативним вимогам кібербезпеки[12]. Для забезпечення цифрової безпеки в системі потрібно враховувати такі моменти:

- 1. Автентифікація та авторизація** (надійні методи автентифікації користувачів, рольова система доступу до функціоналу та даних).
- 2. Шифрування даних** (використання шифрування для зберігання та передачі даних, захист від несанкціонованого доступу).
- 3. Резервне копіювання** (регулярне резервне копіювання даних, можливість швидкого відновлення у разі збоїв або втрати даних).
- 4. Відстеження активності** (занесення в журнал дії користувачів, відстеження підозрілої активності та своєчасне реагування на потенційні загрози).

Забезпечення високого рівня безпеки та захисту даних в CRM-системі є не лише технічним зобов'язанням, але й стратегічною необхідністю для будь-якої компанії, яка прагне захистити свої активи та підтримувати довіру своїх клієнтів.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ

3.1. Вибір мови програмування та фреймворку

Для розробки CRM-системи для малого та середнього бізнесу було проаналізовано багато різноманітних мов програмування та фреймворків. На основі цього аналізу та порівняння можливостей роботи, для створення програмного забезпечення було обрано мову програмування Python та фреймворк Django.

3.1.1 Python

Python[13] — це універсальна мова програмування високого рівня, яка набула широкої популярності завдяки своєму зрозумілому синтаксису, великим можливостям розробки та багатофункціональності. Мову програмування Python використовують у різноманітних ІТ сферах, від веб-розробки та машинного навчання до аналітики даних та створення штучного інтелекту.

Переваги Python:

1. **Універсальність.** Python є універсальною мовою програмування, яку можна використовуватися для різноманітних задач, починаючи від веб-розробок, закінчуючи штучним інтелектом та аналітичним програмним забезпеченням
2. **Простота вивчення.** Простота синтаксису та читабельність коду, робить цю мову чудовим вибором для розробки системи. Завдяки своїй простоті та зрозумілості Python дозволяє максимально оптимізувати код та забезпечити швидкодію системи.
3. **Широкий спектр бібліотек.** Через те, що мова програмування Python є найпопулярнішою мовою програмування в світі, вона має величезну підтримку та велику кількість бібліотек, котрі ідеально підходять для будь яких задач при створенні програмного забезпечення.

3.2 Клієнтська частина

Для створення візуальної частини CRM-системи було обрано мову розмітки HTML, та мову стилю CSS, щоб створити красивий, інтуїтивно зрозумілий та сучасний інтерфейс.

3.2.1 HTML

HTML (HyperText Markup Language) — це мова розмітки, яка використовується для створення базової структури та розмітки сторінок CRM-системи. Завдяки HTML створюється визначення елементів інтерфейсу, таких як заголовки, текстові поля, кнопки та списки.

Використання семантичного HTML5 дозволяє чітко описувати структуру сторінок, що полегшить їх індексацію пошуковими системами та покращить зручність використання.

Для забезпечення адаптивного дизайну, який буде добре виглядати на різних пристроях, буде використовуватися HTML5 з фреймворками, такими як Bootstrap, щоб системою можна було користуватись як на персональному комп'ютері, так і на телефоні, або планшеті.

3.2.2 CSS

CSS (Cascading Style Sheets) — це мова стилю, котра буде використовуватися для стилізації інтерфейсу CRM-системи, роблячи його візуально красивим, сучасним та зручним в користуванні. За допомогою CSS можна зробити налаштування кольорів, шрифтів, розмірів елементів, розкладки та інших різноманітних візуальних елементів.

Використання препроцесорів CSS, таких як Sass, покращить читабельність та організацію коду CSS.

Респонсивна CSS-техніка та фреймворк, такий як Bootstrap Grid, буде використовуватися для створення адаптивного дизайну, який буде добре виглядати на будь-якому екрані, від персональних комп'ютерів до мобільних пристроїв.

3.3 Серверна частина

3.3.1 Django

Django[14] — це високорівневий веб-фреймворк з відкритим кодом, написаний на мові програмування Python. Він використовується для розробки динамічних веб-сайтів та веб-додатків, які є безпечними та простими у використанні. Django був обраний через простоту використання та велику кількість вбудованих додаткових функцій, які можуть в процесі створення програмного забезпечення сильно розширити можливості розробки системи, що позитивно вплине на зручність користування та стабільну роботу системи.

Переваги Django:

- 1. Швидка розробка.** А саме, Django використовує принцип "batteries included" (все включено), що означає, що він поставляється з великою кількістю вбудованих функцій, які часто потрібні для розробки веб-сайтів, таких як автентифікація користувачів, адміністрування контентом та управління URL-адресами.
- 2. Навантаження.** Фреймворк Django побудований на архітектурі MVC (Model-View-Controller), яка відокремлює логіку даних (модель), візуальне представлення (вид) та взаємодію користувача (контролер). Ця архітектура робить Django стійким до навантаження, що дозволяє йому підтримувати веб-сайти та веб-додатки з великою кількістю трафіку та користувачів.
- 3. Безпека.** Мова програмування Python відома своєю можливістю забезпечити захист будь-якого програмного забезпечення, тому і фреймворк Django надає вбудовані функції безпеки, які допомагають захистити веб-сайти та веб-додатки від багатьох популярних веб-загроз.

4. Універсальність. Фреймворк Django настільки універсальний інструмент, що за допомогою нього можна створювати розробки широкого спектру, від веб-сайтів та веб-додатків, до складних інтернет-магазинів та соціальних мереж.

3.4 База даних

Зберігання даних є однією з найважливіших складових будь-якої CRM-системи[12].

CRM-системи збирають, обробляють та зберігають великі обсяги даних про клієнтів, угоди, продажі та інші аспекти бізнесу. Ці дані використовуються для покращення обслуговування клієнтів, підвищення ефективності продажів, прийняття кращих управлінських рішень та багато інших функцій. Вибір відповідної до вимог програмного забезпечення бази даних для зберігання даних CRM-системи є критично важливим. База даних повинна бути локальною, надійною та мати можливість працювати з великим обсягом даних, безпечною та простою у використанні[12].

3.4.1 SQLite

SQLite[15] — це легка, вбудована реляційна база даних, що використовується у багатьох додатках і системах завдяки своїй простоті, надійності та ефективності. Відмінною рисою SQLite є її архітектура: на відміну від традиційних систем управління базами даних, SQLite не потребує окремого сервера для роботи з даними, що робить її ідеальним рішенням для вбудованих систем і додатків з обмеженими ресурсами.

Переваги SQLite:

- 1. Легкість.** SQLite не потребує налаштування сервера, що робить її зручною для розробки та тестування.
- 2. Інтеграція.** База даних SQLite підтримується фреймворком Django по стандарту, не потребуючи додаткової конфігурації та завантаження.
- 3. Зручність.** SQLite має простий та зрозумілий API, що робить її легкою у розробці системи.

- 4. Швидкість.** SQLite є швидкою базою даних, що робить її ідеальною для створення CRM-системи, це позитивно вплине на досвід користування і дозволить користувачам швидко взаємодіяти з клієнтською базою.
- 5. Навантаження.** SQLite може опрацьовувати великі обсяги даних, що є критично важливим аспектом під час росту бізнесу та зростанню кількості клієнтів.
- 6. Безпека.** БД SQLite вміщує в собі ряд функцій безпеки, які потрібно використовувати для захисту даних CRM-системи.

3.5 Проектування CRM-системи

Проектування CRM-системи є комплексним процесом, який охоплює багато етапів та аспектів розробки, щоб створити ефективне та надійне рішення для управління взаємодією з клієнтами[16]. Проектування включає в себе визначення вимог (як функціональних так і не функціональних), розробку архітектури системи, вибір технологій розробки та додаткових інструментів для створення програмного забезпечення.

3.2.1 Архітектура системи

Архітектура CRM-системи є основою її функціональності та продуктивності. Вона визначає, як компоненти системи взаємодіють між собою, як дані зберігаються та обробляються, а також як забезпечується можливість роботи з великим обсягом даних та надійність системи. Розробка архітектури CRM-системи включає вибір архітектурного підходу, розробку модульної структури, інтеграцію з іншими системами та забезпечення безпеки.

Архітектура CRM-системи побудована за принципом MVC. Архітектурна модель MVC (Model-View-Controller) є однією з найбільш поширених архітектурних моделей для розробки програмного забезпечення[16]. Ця модель розділяє додаток на три основні компоненти: модель, вигляд, та контролер. Таке розділення сприяє кращій

організації коду, спрощує підтримку і розвиток додатка, а також дозволяє більш ефективно розробляти взаємодію даних системи.

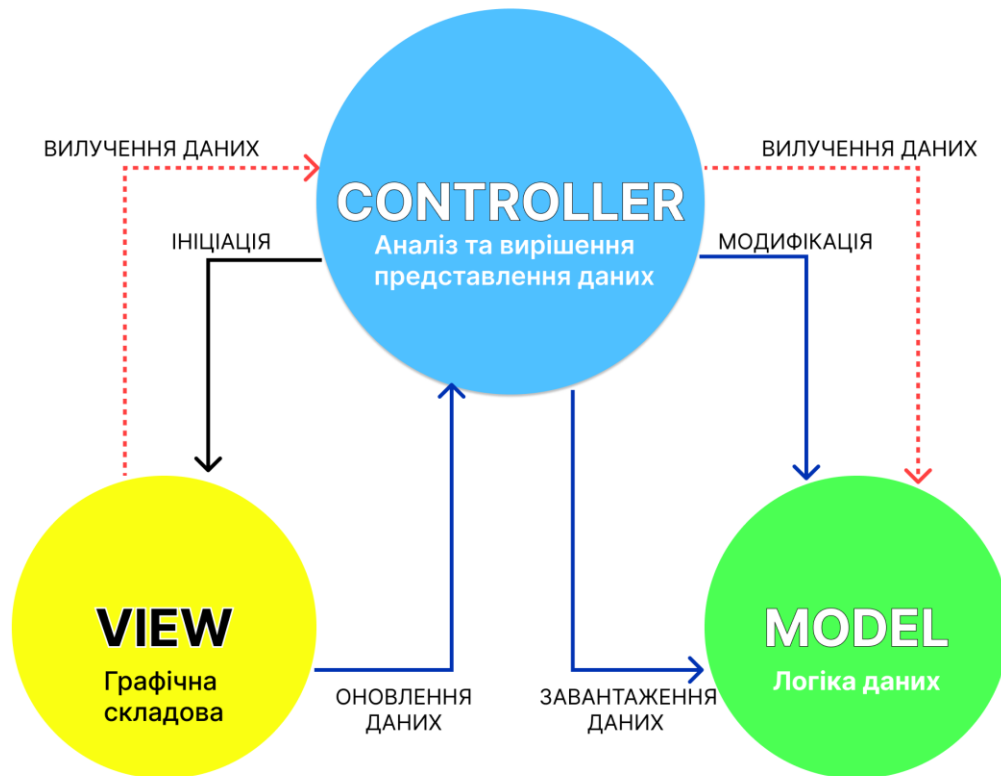


Рис 3.1— Архітектурна модель MVC

- Модель (Model): визначає структуру даних та взаємодію з базою даних.
- Вид (View): відповідає за відображення даних та інтерфейс користувача.
- Контролер (Controller): обробляє запити користувачів, взаємодіє з моделями та визначає, які дані відобразити у видах.

3.6 Основні компоненти системи

Основні компоненти CRM-системи забезпечують ефективне управління взаємодією з клієнтами, автоматизацію процесів та підвищення продуктивності бізнесу. Кожен компонент виконує ключові функції,

необхідні для підтримки і покращення бізнес-процесів. Основні компоненти системи включають[16]:

- **Управління користувачами**

1. Реєстрація користувачів. (Процес створення нових облікових записів для співробітників або адміністраторів системи)
2. Автентифікація та авторизація. (Перевірка особи користувача при вході в систему та надання доступу до відповідних функцій та даних відповідно до його ролей)
3. Управління ролями та правами доступу. (Налаштування ролей (наприклад, адміністратор, менеджер з продажу, підтримка клієнтів) і визначення прав доступу до різних функцій та модулів системи)

- **Управління контактами**

1. Створення контактів. (Додавання нових клієнтів з усією необхідною інформацією, такою як ім'я, адреса, телефон, електронна пошта)
2. Редагування та видалення. (Оновлення інформації про існуючих клієнтів та видалення непотрібних записів)
3. Перегляд інформації (Доступ до детальної інформації про кожного клієнта, включаючи історію взаємодій, попередні угоди та запити)

- **Управління угодами**

1. Управління лідами (Збір та організація потенційних клієнтів, відстеження їхньої активності та інтересів)
2. Відстеження можливостей (Визначення та оцінка можливостей продажу, відстеження прогресу по кожній можливості)
3. Закриття угод (Управління процесом завершення угоди, включаючи узгодження умов, підписання контрактів та обробку платежів)

- **Автоматизація маркетингу**

1. Управління маркетинговими кампаніями (Планування, запуск та відстеження ефективності маркетингових заходів)

2. Сегментація клієнтів (Розподіл клієнтів на різні групи на основі певних критеріїв для цільового маркетингу)
3. Автоматизація комунікацій (Налаштування автоматичних повідомлень, електронних листів та інших видів комунікацій для взаємодії з клієнтами)

- **Обслуговування клієнтів**

1. Обробка запитів (Збір та обробка запитів клієнтів через різні канали, такі як телефон, електронна пошта, чат)
2. Підтримка (Створення та відстеження квитків підтримки, призначення їх відповідним співробітникам та контроль за вирішенням)
3. База знань (Надання клієнтам доступу до статей, інструкцій та відповідей на часто запитувані питання для самостійного вирішення проблем)

- **Аналітика та звітність**

1. Створення звітів (Генерація звітів на основі різних параметрів та показників, такі як продажі, маркетингові кампанії та підтримка)
2. Візуалізація даних (Використання графіків, діаграм та інших візуальних засобів для представлення даних у зрозумілому вигляді)
3. Аналіз ефективності (Оцінка результатів діяльності компанії та окремих підрозділів, виявлення сильних та слабких сторін, прогнозування майбутніх тенденцій)

Кожен з цих компонентів злагоджено працює в системі з іншими, забезпечуючи комплексний підхід до управління взаємодією з клієнтами. Ефективне впровадження та налаштування кожного з цих елементів сприяє підвищенню продуктивності бізнесу, поліпшенню обслуговування клієнтів та збільшенню доходів.

3.7 Розробка модулів CRM-системи

Розробка модулів CRM-системи є важливим етапом створення комплексного рішення для управління взаємодією з клієнтами[17]. Цей

процес включає планування, проєктування та реалізацію різних компонентів системи, що забезпечують функціональність для задоволення потреб бізнесу.

Основні модулі CRM-системи:

- Модуль керування користувачами
- Модуль керування контактами
- Модуль керування угодами
- Модуль керування маркетингу
- Модуль керування клієнтів
- Модуль керування та звітності

3.7.1 Модуль керування користувачами

Модуль керування користувачами є одним із ключових компонентів CRM-системи. Він забезпечує функціонал для реєстрації, автентифікації, авторизації користувачів та управління їхніми ролями і правами доступу. Мета цього модуля — гарантувати, що лише авторизовані користувачі мають доступ до відповідних функцій та даних системи, а також забезпечити зручність і безпеку при роботі з CRM[17].

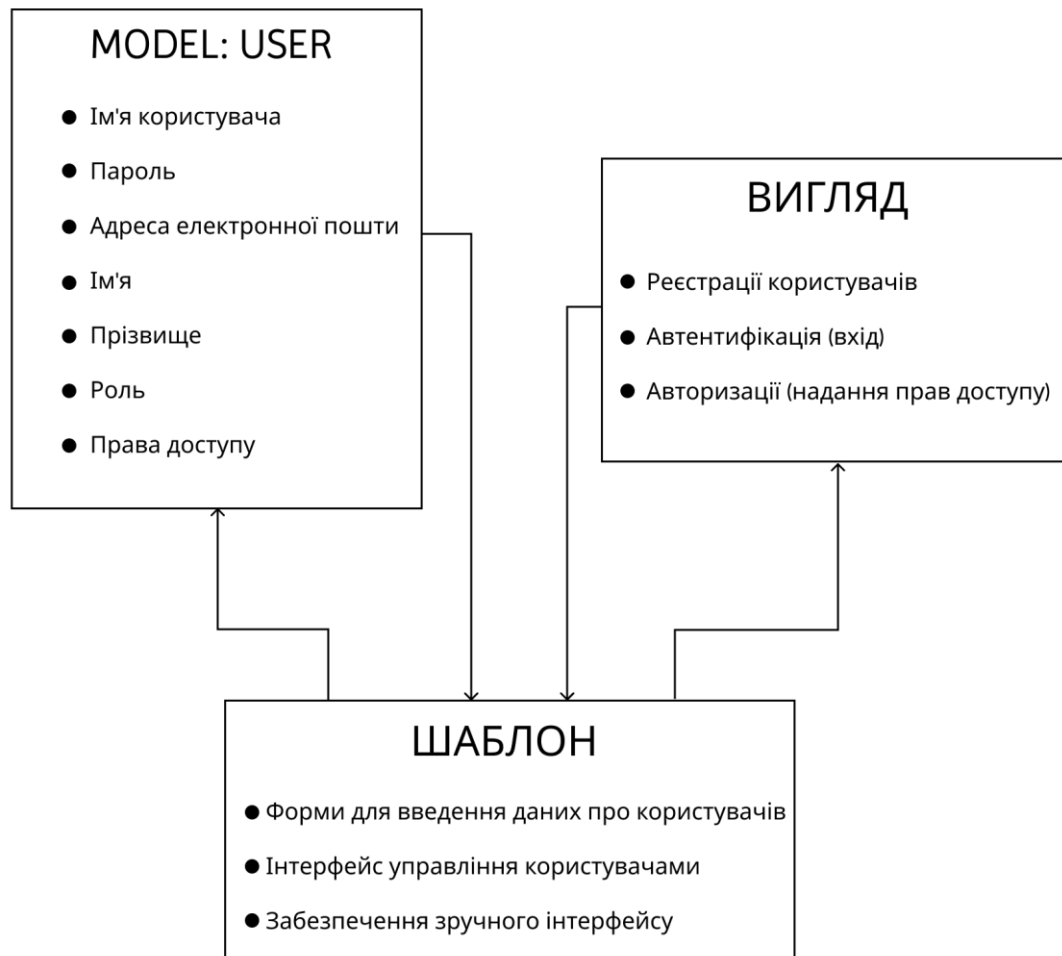


Рис 3.2 — Модуль керування користувачами

- Модель User: визначає структуру даних про користувачів, включаючи інформацію про ролі та права доступу.
- Вигляд: забезпечують функціональність реєстрації, аутентифікації та авторизації.
- Шаблони: форми для введення даних про користувачів, інтерфейс управління користувачами.

3.7.2 Модуль керування контактами

Модуль керування контактами є одним із ключових компонентів будь-якої CRM-системи, оскільки він забезпечує зберігання, організацію та управління всією інформацією про клієнтів[17]. Цей модуль сприяє ефективній взаємодії з клієнтами та допомагає підтримувати високий рівень обслуговування.

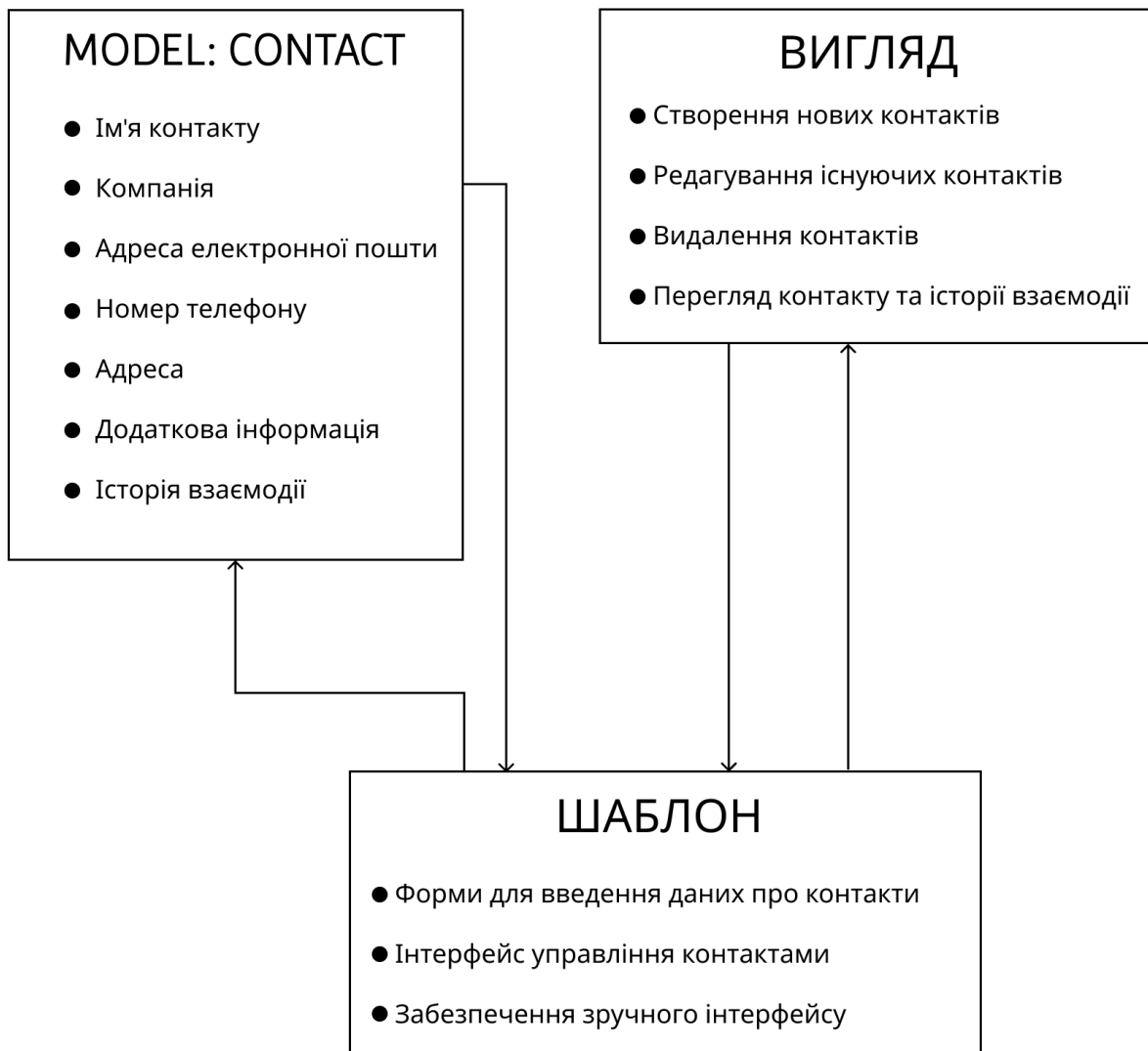


Рис. 3.3 — Модуль керування контактами

- Модель Contact: визначає структуру даних про клієнтів, включаючи контактну інформацію та історію взаємодії.
- Вигляд: забезпечують функціональність створення, редагування, видалення та перегляду контактів.
- Шаблони: форми для введення даних про контакти, інтерфейс управління контактами.

3.7.3 Модуль керування угодами

Модуль керування угодами є важливим CRM-системи, який допомагає відстежувати та керувати всім процесом продажів від початкового контакту до завершення угоди[17]. Цей модуль забезпечує ефективне управління комерційною діяльністю, сприяє підвищенню продуктивності продажів і забезпечує стабільність всіх етапів роботи з клієнтами.

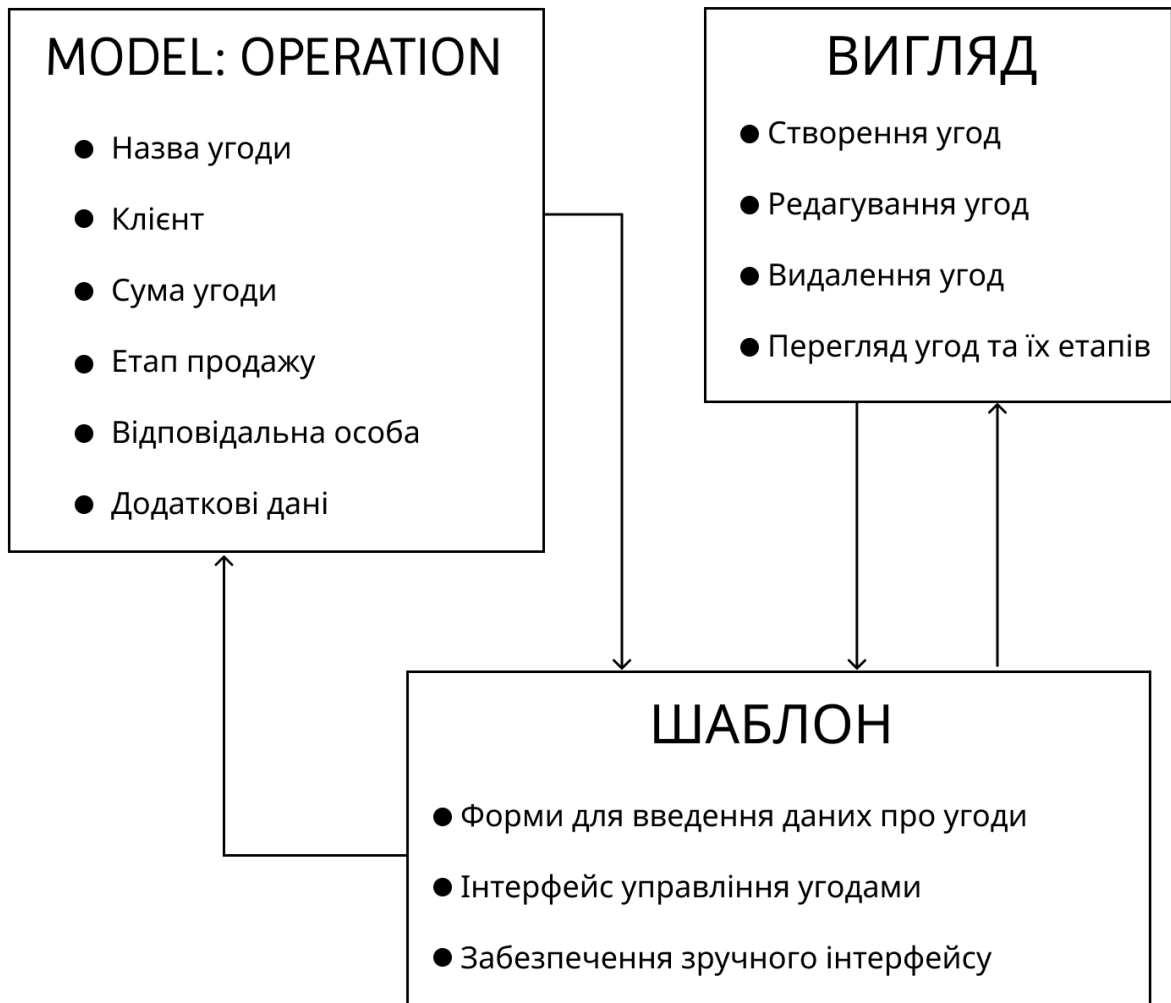


Рис. 3.4 — Модуль керування угодами

- Модель Deal: визначає структуру даних про угоди, включаючи інформацію про етапи продажів та відповідальних осіб.
- Вигляд: забезпечують функціональність створення, редагування, видалення та перегляду угод.

- Шаблони: форми для введення даних про угоди, інтерфейс управління угодами.

3.7.4 Модуль автоматизації маркетингу

Модуль автоматизації маркетингу є складовою маркетингу CRM-системи, що сприяє ефективному проведенню маркетингових кампаній, сегментації клієнтів та аналізу результатів. Він дозволяє автоматизувати маркетингові процеси, зменшуючи потребу в ручній роботі, і забезпечує цілісний підхід до управління взаємодією з клієнтами[17].

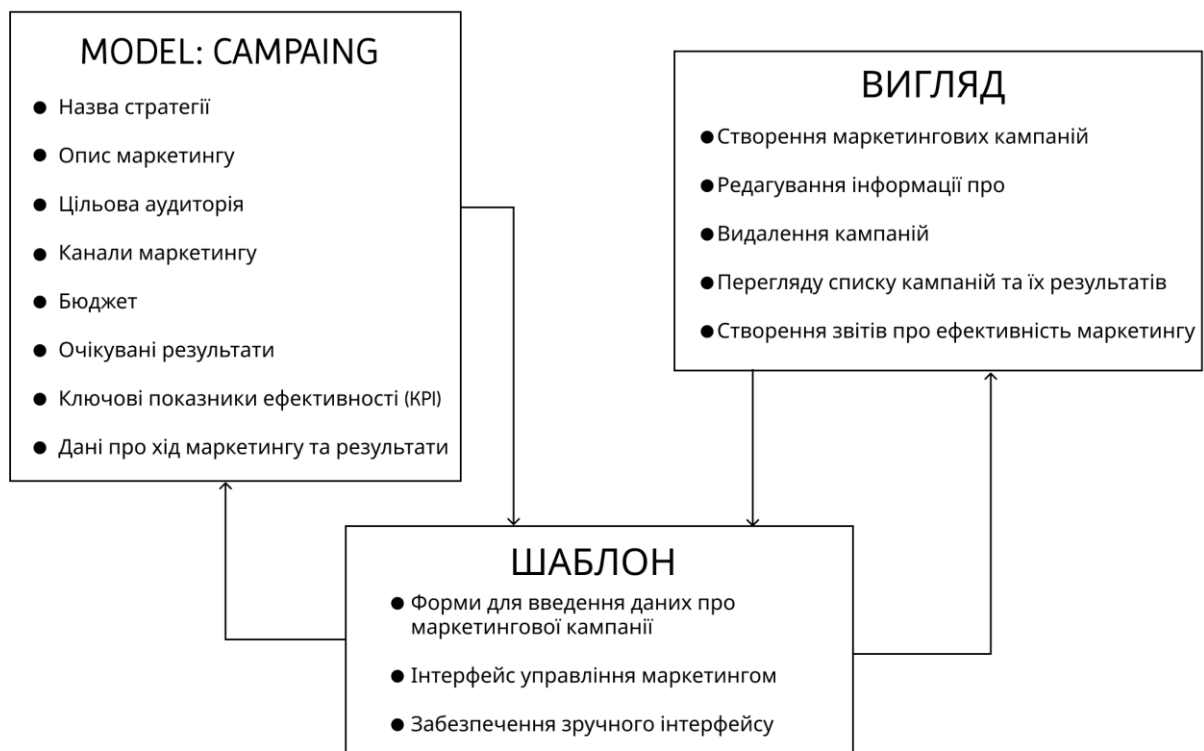


Рис. 3.5 — Модуль автоматизації маркетингу

- Модель Campaign: визначає структуру даних про маркетингові кампанії, включаючи цільову аудиторію та результати.
- Вигляд: забезпечують функціональність створення, редагування, видалення та перегляду кампаній.
- Шаблони: форми для введення даних про кампанії, інтерфейс управління маркетингом.

3.7.5 Модуль обслуговування клієнтів

Модуль обслуговування клієнтів є складовою CRM-системи, який сприяє покращенню якості взаємодії з клієнтами та забезпечує оперативне вирішення їхніх запитів і проблем[17]. Цей модуль допомагає автоматизувати процеси підтримки клієнтів, підвищуючи рівень задоволеності та лояльності клієнтів.

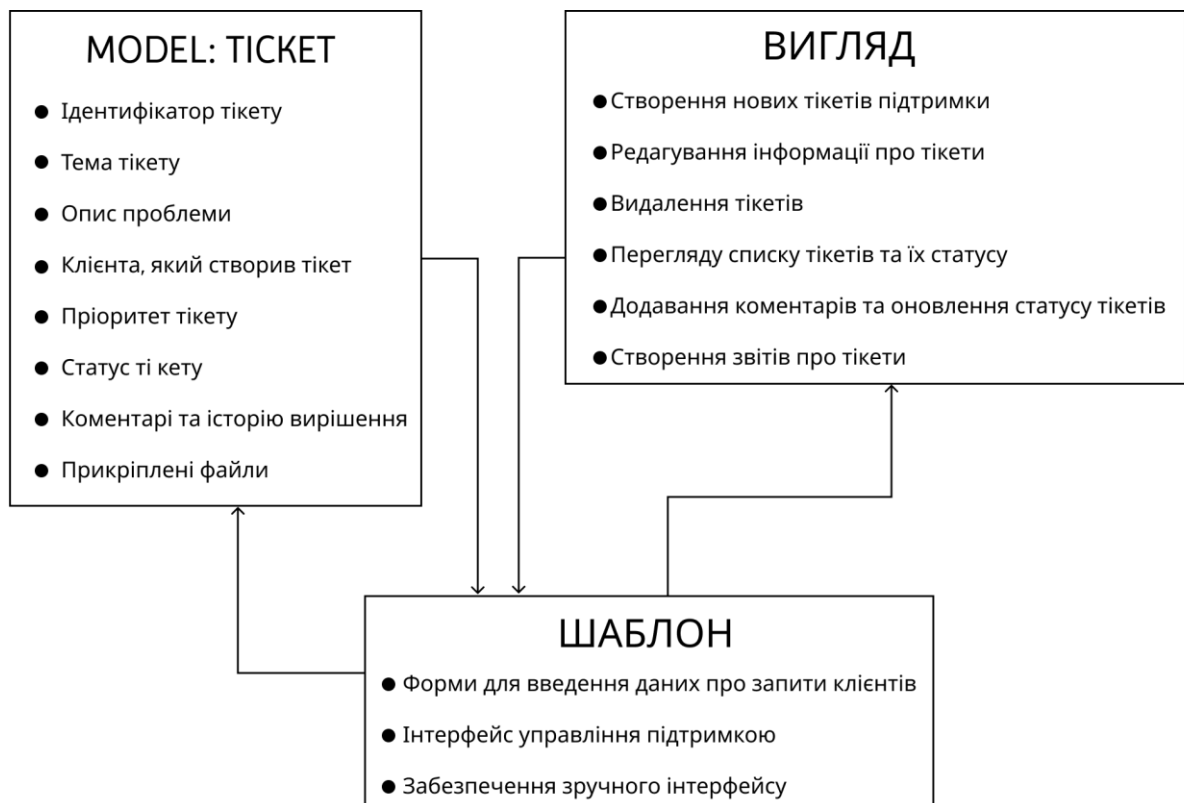


Рис. 3.6 — Модуль обслуговування клієнтів

- **Модель Ticket:** визначає структуру даних про запити клієнтів, включаючи інформацію про проблему та статус вирішення.
- **Вигляд:** забезпечують функціональність створення, редагування, видалення та перегляду квитків підтримки.
- **Шаблони:** форми для введення даних про запити, інтерфейс управління підтримкою.

3.7.6 Модуль аналітики та звітності

Модуль аналітики та звітності допомагає організаціям аналізувати дані, оцінювати ефективність бізнес-процесів і приймати обґрунтовані рішення. Цей модуль забезпечує збирання, обробку та візуалізацію даних, надаючи корисну інформацію для стратегічного планування та оптимізації операцій[17].

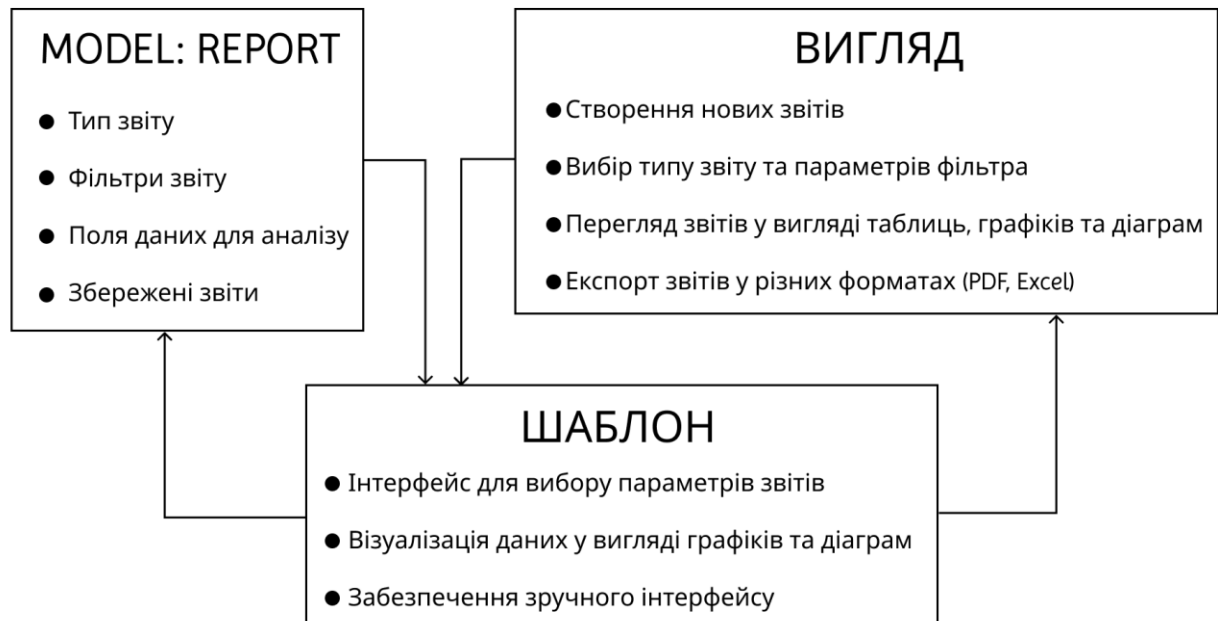


Рис. 3.7 — Модуль аналітики та звітності

- Модель Report: визначає структуру даних про звіти, включаючи інформацію про тип звіту та дані для аналізу.
- Вигляд: забезпечують функціональність створення та перегляду звітів.
- Шаблони: інтерфейс для вибору параметрів звітів, візуалізація даних у вигляді графіків та діаграм.

Розробка CRM-системи за допомогою Python та Django включає вибір відповідних технологій, проектування архітектури системи та створення основних модулів. Дотримання визначених функціональних та нефункціональних вимог забезпечить ефективність та надійність системи, яка задовольнить потреби малого та середнього бізнесу у керуванні взаємодією з клієнтами.

РОЗДІЛ 4. ТЕСТУВАННЯ СИСТЕМИ

З метою підтвердження відповідності розробленої CRM-системи очікуванням та гарантування її безперебійної роботи було проведено комплексне тестування, яке охопило три ключові аспекти: функціональність, продуктивність та зручність користування.

Тестування є невід'ємною частиною процесу розробки програмного забезпечення, особливо для таких критичних систем, як CRM. Метою тестування є виявлення помилок, забезпечення відповідності функціональним і нефункціональним вимогам та підтвердження стабільності й надійності системи.

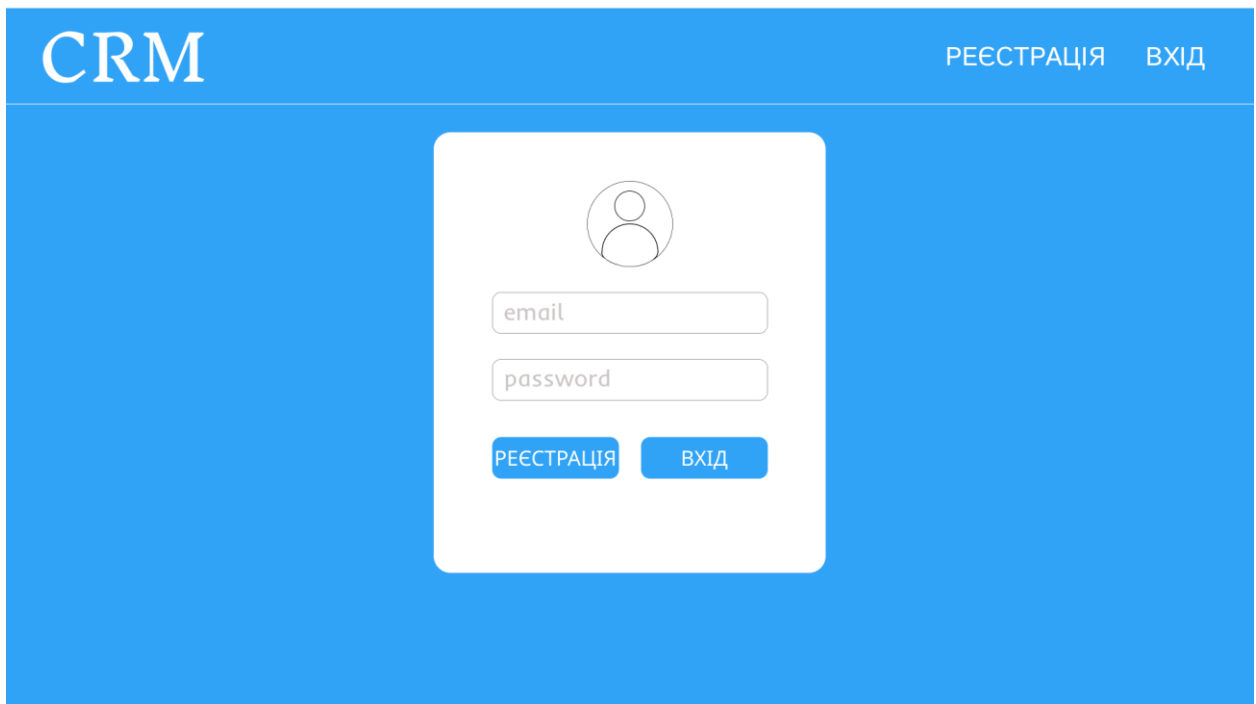


Рис. 4.1 — Вікно входу та реєстрації системи

На початковому етапі тестування було створено базову інфраструктуру CRM, що включає всі ключові модулі: керування користувачами, контактами, угодами, маркетинговими кампаніями, обслуговуванням клієнтів та аналітикою. Кожен модуль пройшов ретельне налаштування та інтеграцію для забезпечення сумісності та безперебійної роботи.

CRM Менеджер

ПРОСТРОЧЕНІ

- Лід: С
Клієнт: Олексій Пономаренко
Тел: 0673125896
Причина: не відповідає

В РОБОТІ

- Лід: В
Клієнт: Андрій Гордієнко
Тел: 0997543359
Замовлення: Xiaomi Mi Power Bank 3
- Лід: А
Клієнт: Валентин Петренко
Тел: 380984551125
Замовлення: Перехідник Promate Link-CI
- Лід: А
Клієнт: Марк Приходько
Тел: 0994532218
Замовлення: Адаптер STLab USB Type-C
- Лід: В
Клієнт: Анастасія
Тел: 0995437781
Замовлення: Кабель Baseus Crystal Shine

НОВІ

- Лід:
Клієнт: Матвій Сергієнко
Тел: 0991524482
Замовлення: Зарядна станція EcoFlow
- Лід:
Клієнт: Валерій Птушкін
Тел: 0974523125
Замовлення: УМБ Duracell CHARGE 10
- Лід:
Клієнт: Вероніка Власова
Тел: 380664279942
Замовлення: Навушники Apple AirPods Pro

Заявки
Замовлення
Контакти
Чат
Пошта
Аналітика

Рис. 4.2 — Вкладка «Заявки»

Окремо було досліджено зручність користування системою.

Оцінювалися інтерфейси користувача, логіка роботи модулів, швидкість завантаження та реакції системи на запити. Отримані дані використовувалися для вдосконалення та доопрацювання системи.

CRM Менеджер

+ СТВОРИТИ ЗАМОВЛЕННЯ

Знайти

| | Дата | Ім'я | Прізвище | Номер телефону | Товар | Сума | Оплата | Статус |
|-----------------------|------------|-----------|-------------|----------------|-------------------|--------|-------------|---------|
| <input type="radio"/> | 14.05.2024 | Матвій | Сергієнко | 0991524482 | Зарядна станці... | 10 000 | Післяоплата | Нові |
| <input type="radio"/> | 14.05.2024 | Валерій | Птушкін | 0974523125 | УМБ Duracell C... | 1 000 | Оплачено | Нові |
| <input type="radio"/> | 14.05.2024 | Вероніка | Власова | 380664279942 | Навушники Арл... | 7 500 | Післяоплата | Нові |
| <input type="radio"/> | 13.05.2024 | Андрій | Гордієнко | 0997543359 | Хіаомі Мі Powe... | 1 125 | Оплачено | Обробка |
| <input type="radio"/> | 13.05.2024 | Валентин | Петренко | 380984551125 | Перехідник Про... | 215 | Післяоплата | Обробка |
| <input type="radio"/> | 13.05.2024 | Марк | Приходько | 0994532218 | Адаптер STLab... | 159 | Післяоплата | Обробка |
| <input type="radio"/> | 13.05.2024 | Анастасія | | 0995437781 | Кабель Baseus... | 175 | Післяоплата | Обробка |
| <input type="radio"/> | 12.05.2024 | Олексій | Пономаренко | 0975431218 | Навушники Raz... | 7300 | Не оплачено | Відміна |
| <input type="radio"/> | 09.05.2024 | Данііл | Васюк | 0665421158 | Зарядна станці... | 68 500 | Післяоплата | Успішно |
| <input type="radio"/> | 07.05.2024 | Кирило | Дмитренко | 0994531257 | УМБ XO Power ... | 1 890 | Оплачено | Успішно |

Заявки
Замовлення
Контакти
Чат
Пошта
Аналітика

Рис. 4.3 — Вкладка «Замовлення»

Наступним кроком стала інтеграція CRM-системи з існуючими бізнес-процесами компанії. Цей процес включав налаштування інтерфейсів для імпорту та експорту даних, визначення прав доступу користувачів та підключення до інших корпоративних додатків.

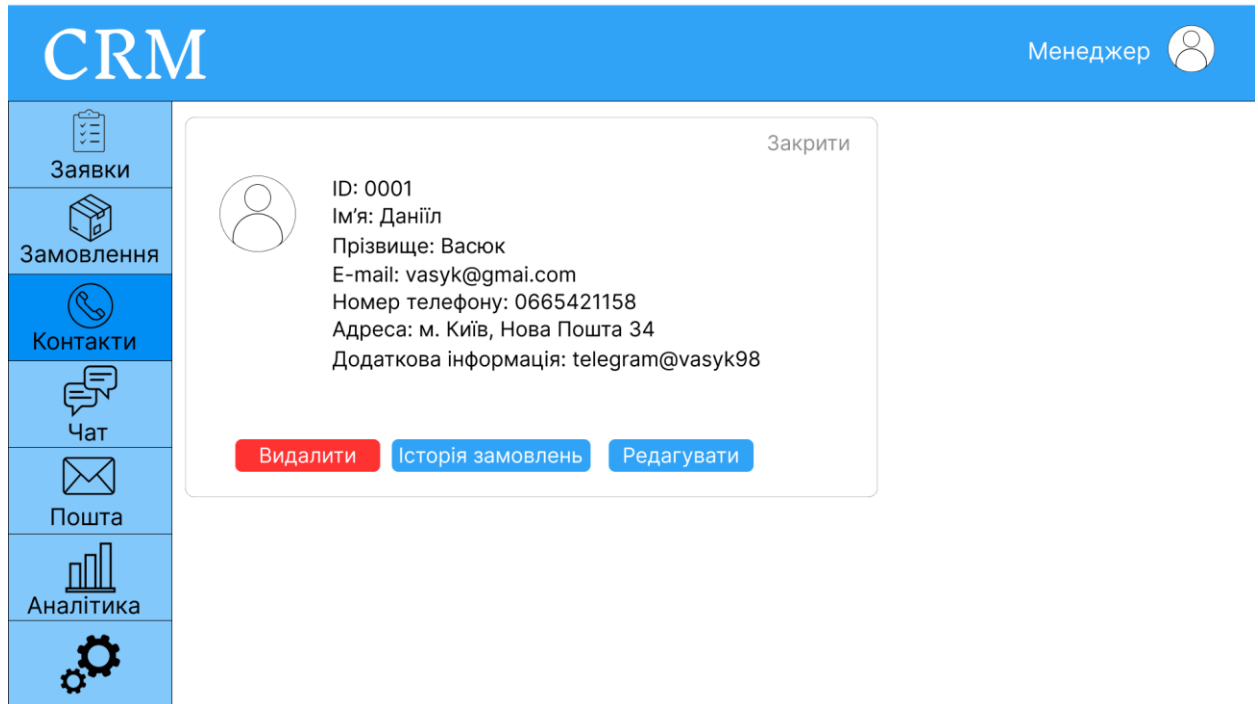


Рис. 4.4 — Вкладка «Контакти»

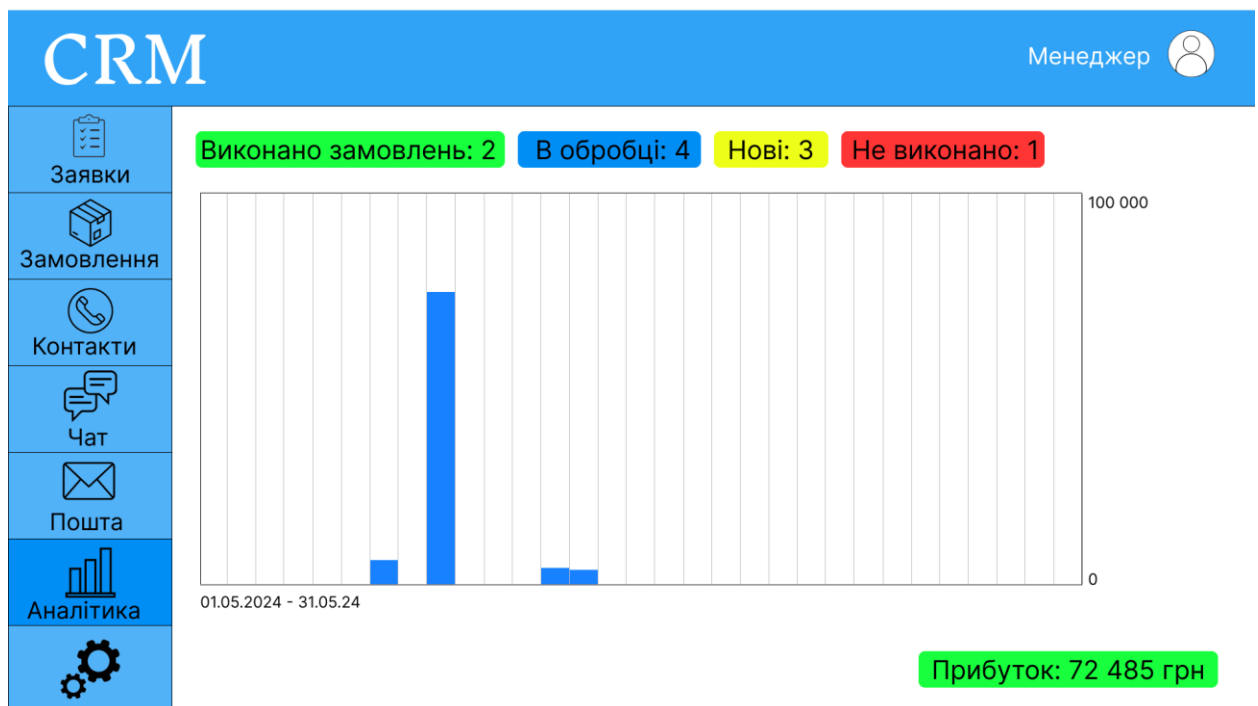


Рис. 4.5 — Вкладка «Аналітика»

Загалом, тестування підтвердило, що CRM-система відповідає всім поставленим вимогам. Вона сприяє ефективному управлінню взаємодією з клієнтами, оптимізує бізнес-процеси та забезпечує високий рівень задоволеності користувачів.

Загальний результат тестування показав, що CRM-система відповідає всім вимогам, забезпечує ефективне управління взаємодією з клієнтами, підвищує продуктивність бізнес-процесів і забезпечує високий рівень задоволення користувачів.

ВИСНОВОК

Розробка та впровадження CRM-системи для малого та середнього бізнесу на базі Python та Django дозволяє значно покращити процеси управління взаємодією з клієнтами, що є критично важливим для успішного функціонування сучасного бізнесу. У ході дослідження було досягнуто наступних результатів:

Огляд предметної області: Було проведено детальний аналіз поняття CRM-систем, їх функціональних можливостей та класифікації. Також було здійснено огляд існуючих рішень на ринку, що дозволило визначити основні переваги та недоліки поточних систем.

Визначення функціональних вимог: На основі аналізу було визначено основні функціональні та нефункціональні вимоги до CRM-системи. Це включає управління контактами, угодами, автоматизацію маркетингу, обслуговування клієнтів, аналітику та звітність, а також забезпечення безпеки, продуктивності, масштабованості та зручності використання.

Розробка системи: Використовуючи Python та Django, було розроблено CRM-систему, що відповідає визначеним вимогам. Проектування системи базувалося на архітектурі MVC, а клієнтська частина була реалізована за допомогою HTML. Всі основні модулі системи, включаючи управління користувачами, контактами, угодами, маркетингом та обслуговуванням клієнтів, були успішно розроблені.

Апробація системи: Проведена апробація системи дозволила оцінити її функціональність, надійність та зручність використання в реальних умовах. Отримані результати показали високу відповідність системи встановленим вимогам, ефективність її роботи та позитивні відгуки від тестових користувачів.

Оптимізація та вдосконалення: На основі зворотного зв'язку було здійснено виправлення виявлених помилок та оптимізація системи. Це

включало покращення продуктивності, зручності користування та безпеки системи.

В результаті, розроблена CRM-система на базі Python та Django є ефективним інструментом для малого та середнього бізнесу, що дозволяє підвищити якість управління взаємодією з клієнтами, оптимізувати бізнес-процеси та сприяти зростанню компаній. Впровадження такої системи забезпечить покращення обслуговування клієнтів, збільшення продажів та ефективне управління маркетинговими кампаніями.

Подальші кроки передбачають повноцінне впровадження системи в реальні бізнес-процеси, подальшу підтримку та оновлення для забезпечення актуальності та відповідності сучасним вимогам ринку.

СПИСОК ЛІТЕРАТУРИ

1. CRM-системи в малому та середньому бізнесі: теорія і практика. Черненко С. М. – Київ: Видавництво "Ліра-К", 2018. 300 – 320 с.
2. Автоматизація бізнес-процесів з використанням CRM-систем. Петрович І. В. – Львів: Видавництво "Магнолія-2006", 2020. 175 – 280 с.
3. Customer Relationship Management: Concepts and Technologies / Anderson, J. R. – New York: Springer, 2015. 48 – 63 с.
4. Офіційний сайт Salesforce — URL: <https://www.salesforce.com/>
5. Офіційний сайт HubSpot CRM — URL: <https://www.hubspot.com/products/crm>
6. Офіційний сайт SalesDrive — URL: <https://salesdrive.ua/>
7. Офіційний сайт KeyCRM — URL: <https://ua.keycrm.app/>
8. Сучасні тенденції розвитку CRM-систем / Коваленко О. П. – Харків: Видавництво "Фоліо", 2017. 125 – 130 с.
9. CRM at the Speed of Light: Social CRM Strategies, Tools, and Techniques for Engaging Your Customers / Greenberg, P. – New York: McGraw-Hill, 2010. 75 – 88 с.
10. Інформаційні системи управління взаємодією з клієнтами / Довгань В. М. – Дніпро: Видавництво "Ліра", 2019. 163 – 185 с.
11. Customer Relationship Management: Concepts and Technologies / Buttle, F. – London: Routledge, 2016. 110 – 147 с.
12. Використання CRM-систем у маркетинговій діяльності підприємств / Соколовська А. В. – Одеса: Видавництво "Одеська національна академія", 2018. 12 – 24 с.
13. Офіційний сайт Python — URL: <https://www.python.org/>
14. Офіційний сайт Django — URL: <https://www.djangoproject.com/>
15. Офіційний сайт SQLite — URL: <https://sqlite.org/>
16. Managing Customer Relationships: A Strategic Framework / Peppers, D., Rogers, M. – Hoboken: John Wiley & Sons, 2011. 96 – 105 с.

17. Розробка та впровадження CRM-систем на підприємствах малого та середнього бізнес. Кравченко І. П. – Київ: Видавництво "Наукова думка", 2021. 34 – 52 с.

ДОДАТОК А

```
from django.db import models
```

```
from django.contrib.auth.models import User
```

```
from accounts.models import Account
```

```
from contacts.models import Contact
```

```
STAGES = (
```

```
    ('QUALIFICATION', 'QUALIFICATION'),
```

```
    ('NEEDS ANALYSIS', 'NEEDS ANALYSIS'),
```

```
    ('VALUE PROPOSITION', 'VALUE PROPOSITION'),
```

```
    ('ID.DECISION MAKERS', 'ID.DECISION MAKERS'),
```

```
    ('PERCEPTION ANALYSIS', 'PERCEPTION ANALYSIS'),
```

```
    ('PROPOSAL/PRICE QUOTE', 'PROPOSAL/PRICE QUOTE'),
```

```
    ('NEGOTIATION/REVIEW', 'NEGOTIATION/REVIEW'),
```

```
    ('CLOSED WON', 'CLOSED WON'),
```

```
    ('CLOSED LOST', 'CLOSED LOST'),
```

```
)
```

```
SOURCES = (
```

```
    ('NONE', 'NONE'),
```

```
    ('CALL', 'CALL'),
```

```
    ('EMAIL', 'EMAIL'),
```

```
    ('EXISTING CUSTOMER', 'EXISTING CUSTOMER'),
```

```
    ('PARTNER', 'PARTNER'),
```

```
    ('PUBLIC RELATIONS', 'PUBLIC RELATIONS'),
```

```
    ('CAMPAIGN', 'CAMPAIGN'),
```

```
    ('WEBSITE', 'WEBSITE'),
```

```
    ('OTHER', 'OTHER'),
```

```
)
```

```
TYPECHOICES = (  
    ('CUSTOMER', 'CUSTOMER'),  
    ('INVESTOR', 'INVESTOR'),  
    ('PARTNER', 'PARTNER'),  
    ('RESELLER', 'RESELLER'),  
)
```

```
INDCHOICES = (  
    ('ADVERTISING', 'ADVERTISING'),  
    ('AGRICULTURE', 'AGRICULTURE'),  
    ('APPAREL & ACCESSORIES', 'APPAREL & ACCESSORIES'),  
    ('AUTOMOTIVE', 'AUTOMOTIVE'),  
    ('BANKING', 'BANKING'),  
    ('BIOTECHNOLOGY', 'BIOTECHNOLOGY'),  
    ('BUILDING MATERIALS & EQUIPMENT', 'BUILDING MATERIALS &  
EQUIPMENT'),  
    ('CHEMICAL', 'CHEMICAL'),  
    ('COMPUTER', 'COMPUTER'),  
    ('EDUCATION', 'EDUCATION'),  
    ('ELECTRONICS', 'ELECTRONICS'),  
    ('ENERGY', 'ENERGY'),  
    ('ENTERTAINMENT & LEISURE', 'ENTERTAINMENT & LEISURE'),  
    ('FINANCE', 'FINANCE'),  
    ('FOOD & BEVERAGE', 'FOOD & BEVERAGE'),  
    ('GROCERY', 'GROCERY'),  
    ('HEALTHCARE', 'HEALTHCARE'),  
    ('INSURANCE', 'INSURANCE'),  
    ('LEGAL', 'LEGAL'),
```

```

('MANUFACTURING', 'MANUFACTURING'),
('PUBLISHING', 'PUBLISHING'),
('REAL ESTATE', 'REAL ESTATE'),
('SERVICE', 'SERVICE'),
('SOFTWARE', 'SOFTWARE'),
('SPORTS', 'SPORTS'),
('TECHNOLOGY', 'TECHNOLOGY'),
('TELECOMMUNICATIONS', 'TELECOMMUNICATIONS'),
('TELEVISION', 'TELEVISION'),
('TRANSPORTATION', 'TRANSPORTATION'),
('VENTURE CAPITAL', 'VENTURE CAPITAL'),
)

```

```

class Opportunity(models.Model):
    name = models.CharField(max_length=64)
    account = models.ForeignKey(Account, related_name='opportunities',
on_delete=models.CASCADE, blank=True, null=True)
    stage = models.CharField( max_length=64, choices=STAGES)
    amount = models.DecimalField("Opportunity Amount", decimal_places=2,
max_digits=12, blank=True, null=True)
    lead_source = models.CharField("Source of Lead", max_length=255,
choices=SOURCES, blank=True, null=True)
    probability = models.IntegerField(default=0, blank=True, null=True)
    contacts = models.ManyToManyField(Contact)
    closedBy = models.ForeignKey(User, on_delete=models.CASCADE,
null=True)
    closedOn = models.DateTimeField(blank=True, null=True)
    description = models.TextField(blank=True, null=True)
    createdBy = models.ForeignKey(User, related_name='opportunity_created_by',
on_delete=models.CASCADE)

```



```

createdOn = models.DateTimeField("Created on", auto_now_add=True)
isActive = models.BooleanField(default=False)

def __str__(self):
    return self.name
class Migration(migrations.Migration):

    initial = True

    dependencies = [
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]

    operations = [
        migrations.CreateModel(
            name='Account',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('Name', models.CharField(max_length=64, verbose_name='Name of
Account')),
                ('email', models.EmailField(max_length=254)),
                ('phone', models.CharField(max_length=20)),
                ('industry', models.CharField(blank=True, choices=[('FINANCE',
'FINANCE'), ('HEALTHCARE', 'HEALTHCARE'), ('INSURANCE',
'INSURANCE'), ('LEGAL', 'LEGAL'), ('MANUFACTURING',
'MANUFACTURING'), ('PUBLISHING', 'PUBLISHING'), ('REAL ESTATE',
'REAL ESTATE'), ('SOFTWARE', 'SOFTWARE')], max_length=255, null=True,
verbose_name='Industry Type')),

```

```

        ('website', models.URLField(blank=True, null=True,
verbose_name='Website')),
        ('description', models.TextField(blank=True, null=True)),
        ('createdOn', models.DateTimeField(auto_now_add=True,
verbose_name='Created on')),
        ('isActive', models.BooleanField(default=False)),
        ('assigned_to',
models.ManyToManyField(related_name='account_assigned_to',
to=settings.AUTH_USER_MODEL)),
        ('createdBy',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_name='account_created_by', to=settings.AUTH_USER_MODEL)),
    ],
),
]

```

```
class Migration(migrations.Migration):
```

```
    initial = True
```

```

dependencies = [
    ('accounts', '0001_initial'),
    migrations.swappable_dependency(settings.AUTH_USER_MODEL),
]

```

```

operations = [
    migrations.CreateModel(
        name='Contact',
        fields=[
            ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),

```

```

        ('first_name', models.CharField(max_length=255, verbose_name='First
name')),
        ('last_name', models.CharField(max_length=255, verbose_name='Last
name')),
        ('email', models.EmailField(max_length=254)),
        ('phone', models.CharField(max_length=20)),
        ('address', models.TextField(blank=True, null=True)),
        ('description', models.TextField(blank=True, null=True)),
        ('createdOn', models.DateTimeField(auto_now_add=True,
verbose_name='Created on')),
        ('isActive', models.BooleanField(default=False)),
        ('account', models.ForeignKey(blank=True, null=True,
on_delete=django.db.models.deletion.CASCADE,
related_name='lead_account_contacts', to='accounts.Account')),
        ('createdBy',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_name='contact_created_by', to=settings.AUTH_USER_MODEL)),
    ],
),
]

```

```
class Contact(models.Model):
```

```

    first_name = models.CharField("First name", max_length=255)
    last_name = models.CharField("Last name", max_length=255)
    account = models.ForeignKey(Account, related_name='lead_account_contacts',
on_delete=models.CASCADE, blank=True, null=True)
    email = models.EmailField()
    phone = models.CharField(max_length=20)
    address = models.TextField(blank=True, null=True)
    description = models.TextField(blank=True, null=True)

```

```

    createdBy = models.ForeignKey(User, related_name='contact_created_by',
on_delete=models.CASCADE)
    createdOn = models.DateTimeField("Created on", auto_now_add=True)
    isActive = models.BooleanField(default=False)

    def __str__(self):
        return self.first_name
class Migration(migrations.Migration):

    initial = True

    dependencies = [
        ('accounts', '0001_initial'),
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]

    operations = [
        migrations.CreateModel(
            name='Lead',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('title', models.CharField(blank=True, max_length=64, null=True,
verbose_name='Treatment Pronouns for the customer')),
                ('first_name', models.CharField(max_length=255, verbose_name='First
name')),
                ('last_name', models.CharField(max_length=255, verbose_name='Last
name')),
                ('email', models.EmailField(max_length=254)),
                ('phone', models.CharField(blank=True, max_length=20, null=True)),

```

```

        ('status', models.CharField(blank=True, choices=[('assigned', 'Assigned'),
('in process', 'In Process'), ('converted', 'Converted'), ('recycled', 'Recycled'), ('dead',
'Dead')], max_length=255, null=True, verbose_name='Status of Lead')),
        ('source', models.CharField(blank=True, choices=[('call', 'Call'), ('email',
'Email'), ('existing customer', 'Existing Customer'), ('partner', 'Partner'), ('public
relations', 'Public Relations'), ('compaign', 'Campaign'), ('other', 'Other')],
max_length=255, null=True, verbose_name='Source of Lead')),
        ('address', models.CharField(blank=True, max_length=255, null=True,
verbose_name='Address')),
        ('website', models.CharField(blank=True, max_length=255, null=True,
verbose_name='Website')),
        ('description', models.TextField(blank=True, null=True)),
        ('account_name', models.CharField(blank=True, max_length=255,
null=True)),
        ('opportunity_amount', models.DecimalField(blank=True,
decimal_places=2, max_digits=12, null=True, verbose_name='Opportunity
Amount')),
        ('createdOn', models.DateTimeField(auto_now_add=True,
verbose_name='Created on')),
        ('isActive', models.BooleanField(default=False)),
        ('enquiry_type', models.CharField(blank=True, max_length=255,
null=True)),
        ('account', models.ForeignKey(blank=True, null=True,
on_delete=django.db.models.deletion.CASCADE, related_name='Leads',
to='accounts.Account')),
        ('assigned_to',
models.ManyToManyField(related_name='lead_assigned_users',
to=settings.AUTH_USER_MODEL)),

```

```

        ('createdBy',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_name='lead_created_by', to=settings.AUTH_USER_MODEL)),
    ],
),
]

```

```

LEAD_SOURCE = (
    ('call', 'Call'),
    ('email', 'Email'),
    ('existing customer', 'Existing Customer'),
    ('partner', 'Partner'),
    ('public relations', 'Public Relations'),
    ('campaign', 'Campaign'),
    ('other', 'Other'),
)

```

```

LEAD_STATUS = (
    ('assigned', 'Assigned'),
    ('in process', 'In Process'),
    ('converted', 'Converted'),
    ('recycled', 'Recycled'),
    ('dead', 'Dead')
)

```

```

class Lead(models.Model):
    title = models.CharField("Treatment Pronouns for the customer",
        max_length=64, blank=True, null=True)
    first_name = models.CharField(("First name"), max_length=255)
    last_name = models.CharField(("Last name"), max_length=255)
    email = models.EmailField()

```

```

    phone = models.CharField(max_length=20, null=True, blank=True)
    account = models.ForeignKey(Account, related_name='Leads',
on_delete=models.CASCADE, blank=True, null=True)
    status = models.CharField("Status of Lead", max_length=255,
        blank=True, null=True, choices=LEAD_STATUS)
    source = models.CharField("Source of Lead", max_length=255,
        blank=True, null=True, choices=LEAD_SOURCE)
    address = models.CharField("Address", max_length=255, blank=True,
null=True)

    website = models.CharField("Website", max_length=255, blank=True,
null=True)
    description = models.TextField(blank=True, null=True)
    assigned_to = models.ManyToManyField(User,
related_name='lead_assigned_users')
    account_name = models.CharField(max_length=255, null=True, blank=True)
    opportunity_amount = models.DecimalField("Opportunity Amount",
decimal_places=2, max_digits=12,
        blank=True, null=True)
    createdBy = models.ForeignKey(User, related_name='lead_created_by',
on_delete=models.CASCADE)
    createdOn = models.DateTimeField("Created on", auto_now_add=True)
    isActive = models.BooleanField(default=False)
    enquiry_type = models.CharField(max_length=255, blank=True, null=True)

    def __str__(self):
        return self.first_name + self.last_name
migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ('accounts', '0001_initial'),
    ('contacts', '0001_initial'),

```

```

]

operations = [
    migrations.CreateModel(
        name='Opportunity',
        fields=[
            ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
            ('name', models.CharField(max_length=64)),
            ('stage', models.CharField(choices=[('QUALIFICATION',
'QUALIFICATION'), ('NEEDS ANALYSIS', 'NEEDS ANALYSIS'), ('VALUE
PROPOSITION', 'VALUE PROPOSITION'), ('ID.DECISION MAKERS',
'ID.DECISION MAKERS'), ('PERCEPTION ANALYSIS', 'PERCEPTION
ANALYSIS'), ('PROPOSAL/PRICE QUOTE', 'PROPOSAL/PRICE QUOTE'),
('NEGOTIATION/REVIEW', 'NEGOTIATION/REVIEW'), ('CLOSED WON',
'CLOSED WON'), ('CLOSED LOST', 'CLOSED LOST')], max_length=64)),
            ('amount', models.DecimalField(blank=True, decimal_places=2,
max_digits=12, null=True, verbose_name='Opportunity Amount')),
            ('lead_source', models.CharField(blank=True, choices=[('NONE',
'NONE'), ('CALL', 'CALL'), ('EMAIL', 'EMAIL'), ('EXISTING CUSTOMER',
'EXISTING CUSTOMER'), ('PARTNER', 'PARTNER'), ('PUBLIC RELATIONS',
'PUBLIC RELATIONS'), ('CAMPAIGN', 'CAMPAIGN'), ('WEBSITE',
'WEBSITE'), ('OTHER', 'OTHER')], max_length=255, null=True,
verbose_name='Source of Lead')),
            ('probability', models.IntegerField(blank=True, default=0, null=True)),
            ('closedOn', models.DateTimeField(blank=True, null=True)),
            ('description', models.TextField(blank=True, null=True)),
            ('createdOn', models.DateTimeField(auto_now_add=True,
verbose_name='Created on')),
            ('isActive', models.BooleanField(default=False)),

```



```

        ('account', models.ForeignKey(blank=True, null=True,
on_delete=django.db.models.deletion.CASCADE, related_name='opportunities',
to='accounts.Account')),
        ('closedBy', models.ForeignKey(null=True,
on_delete=django.db.models.deletion.CASCADE,
to=settings.AUTH_USER_MODEL)),
        ('contacts', models.ManyToManyField(to='contacts.Contact')),
        ('createdBy',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_name='opportunity_created_by', to=settings.AUTH_USER_MODEL)),
    ],
),
] class Migration(migrations.Migration):

    initial = True

    dependencies = [
        ('accounts', '0001_initial'),
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]

    operations = [
        migrations.CreateModel(
            name='Lead',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('title', models.CharField(blank=True, max_length=64, null=True,
verbose_name='Treatment Pronouns for the customer')),

```

```

('first_name', models.CharField(max_length=255, verbose_name='First
name')),
('last_name', models.CharField(max_length=255, verbose_name='Last
name')),
('email', models.EmailField(max_length=254)),
('phone', models.CharField(blank=True, max_length=20, null=True)),
('status', models.CharField(blank=True, choices=[('assigned', 'Assigned'),
('in process', 'In Process'), ('converted', 'Converted'), ('recycled', 'Recycled'), ('dead',
'Dead')], max_length=255, null=True, verbose_name='Status of Lead')),
('source', models.CharField(blank=True, choices=[('call', 'Call'), ('email',
'Email'), ('existing customer', 'Existing Customer'), ('partner', 'Partner'), ('public
relations', 'Public Relations'), ('compaign', 'Campaign'), ('other', 'Other')],
max_length=255, null=True, verbose_name='Source of Lead')),
('address', models.CharField(blank=True, max_length=255, null=True,
verbose_name='Address')),
('website', models.CharField(blank=True, max_length=255, null=True,
verbose_name='Website')),
('description', models.TextField(blank=True, null=True)),
('account_name', models.CharField(blank=True, max_length=255,
null=True)),
('opportunity_amount', models.DecimalField(blank=True,
decimal_places=2, max_digits=12, null=True, verbose_name='Opportunity
Amount')),
('createdOn', models.DateTimeField(auto_now_add=True,
verbose_name='Created on')),
('isActive', models.BooleanField(default=False)),
('enquiry_type', models.CharField(blank=True, max_length=255,
null=True)),

```

```

        ('account', models.ForeignKey(blank=True, null=True,
on_delete=django.db.models.deletion.CASCADE, related_name='Leads',
to='accounts.Account')),
        ('assigned_to',
models.ManyToManyField(related_name='lead_assigned_users',
to=settings.AUTH_USER_MODEL)),
        ('createdBy',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
related_name='lead_created_by', to=settings.AUTH_USER_MODEL)),
    ],
),
]
LIST_HEADERS = (
    (u'Client', 'name'),
)

```

```
@login_required
```

```
def client_list(request):
```

```
    """
```

```
    Lists Clients.
```

```
    """
```

```
    sort_headers = SortHeaders(request, LIST_HEADERS)
```

```
    return list_detail.object_list(request,
```

```
        Client.objects.order_by(sort_headers.get_order_by()),
```

```
        paginate_by=settings.ITEMS_PER_PAGE, allow_empty=True,
```

```
        template_object_name='client',
```

```
        template_name='clients/client_list.html', extra_context={
```

```
            'headers': list(sort_headers.headers()),
```

```
        })
```

```
@login_required
def add_client(request):
    """
    Adds a Client.
    """
    return add_object(request, Client, fields=('name', 'notes'),
        template_name='clients/add_client.html')
```

```
@login_required
def client_detail(request, client_id):

    client = get_object_or_404(Client, pk=client_id)
    jobs = Job.objects.accessible_to_user(request.user) \
        .filter(client=client).order_by('number')
    return render_to_response('clients/client_detail.html', {
        'client': client,
        'jobs': jobs,
        'contacts': client.contacts.all(),
    }, RequestContext(request))
```

```
@login_required
def edit_client(request, client_id):
    """
    Edit a Client.
    """
    return edit_object(request, Client, client_id,
        fields=('name', 'notes', 'disabled'),
        template_name='clients/edit_client.html',
        template_object_name='client')
```

```
@user_has_permission(is_admin_or_manager)
def delete_client(request, client_id):
    """
    Deletes a Client.
    """
    client = get_object_or_404(Client, pk=client_id)
    if not client.is_deleteable:
        return HttpResponseRedirect(u'The selected %s is not deleteable.' \
                                    % Client._meta.verbose_name)
    return create_update.delete_object(request, Client,
        post_delete_redirect=reverse('client_list'), object_id=client_id,
        template_object_name='client',
```