|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Aplikacja CRM | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Specyfikacja Projektowa |  | | | |
|  | | | | \_Bartłomiej Antos |  | | | |
|  | | |  | | |  | | |

Spis treści

1. **Cel i opis projektu** 1
2. **Wymagania klienta** 2
3. **Możliwe analizy danych w aplikacji** 4
4. **Wymagania funkcjonalne** 5
5. **Wymagania niefunkcjonalne** 5
6. **Wykorzystane Technologie**6
7. **Przykładowa architektura bazy danych** 6
8. **Product Backlog** 8
9. **Załączniki** 9
10. **Cel i opis projektu**

Projekt dotyczy stworzenia aplikacji desktopowej CRM (Customer Relationship Management) w języku Python, wykorzystującej bibliotekę PyQ6. Aplikacja będzie umożliwiała zarządzanie danymi związanych z klientami, kontaktami, możliwościami, ofertami, zamówieniami oraz produktami. Głównym celem aplikacji będzie umożliwienie użytkownikom przeglądania, dodawania, edytowania i usuwania danych w bazie, a także generowanie analiz i wykresów na podstawie posiadanych danych.

1. **Wymagania klienta**

* Intuicyjny interfejs użytkownika.
* Aplikacja powinna umożliwiać operacje CRUD dla obiektów: Accounts, Contacts, Opportunity, Quote, Order, Product.
* Obsługa różnych typów użytkowników.
* Personalizacja widoków i preferencji.
* Bezpieczeństwo danych.
* Wczytywanie danych z plików CSV z separatorem w postaci przecinka.
* Analityka sprzedażowa i generowanie raportów.
* Wydajność i skalowalność.
* Powinna być dostępna funkcjonalność tabularycznego podglądu danych dla każdego z obiektów.
* Użytkownicy powinni mieć możliwość załadowania danych z plików CSV.
* Aplikacja powinna korzystać z bibliotek analitycznych języka Python do tworzenia analiz i wykresów na podstawie posiadanych danych.

1. **Możliwe analizy danych w aplikacji**

* **Analiza sprzedaży**: Generowanie raportów i wykresów dotyczących sprzedaży, takich jak zestawienia sprzedaży w czasie, analiza wykresów sprzedaży produktów, analiza sprzedaży według klientów lub regionów.
* **Analiza konwersji**: Śledzenie procesu konwersji klientów od etapu potencjalnej sprzedaży do finalizacji zamówienia. Generowanie raportów i wykresów przedstawiających wskaźniki konwersji, np. odsetek możliwości zamienionych w zamówienia.
* **Analiza klientów**: Tworzenie profili klientów na podstawie zebranych danych i generowanie raportów dotyczących preferencji klientów, historii transakcji, segmentacji klientów według różnych kryteriów.
* **Analiza rentowności**: Obliczanie rentowności klientów, produktów lub określonych transakcji. Identyfikowanie najbardziej dochodowych klientów lub produktów.

Te analizy pomogą użytkownikom w lepszym zrozumieniu danych, podejmowaniu decyzji biznesowych i identyfikowaniu obszarów do poprawy lub rozwoju w ramach zarządzania relacjami z klientem.

1. **Wymagania funkcjonalne**

* Tworzenie, odczyt, aktualizacja i usuwanie obiektów: Accounts, Contacts, Opportunity, Quote, Order, Product.
* Wyświetlanie tabularycznego podglądu danych dla każdego z obiektów.
* Możliwość załadowania danych z plików CSV dla obiektów Accounts, Contacts i Product: obsługiwane operacje Create, Upsert (stworzenie nowych rekordów lub aktualizacja obecnych zlokalizowanych po parametrze ID).
* Implementacja walidacji danych, pozwalająca na zachowanie „czystości” danych.
* Generowanie analiz i wykresów na podstawie dostępnych danych.
* Utworzenie roli administratora, który będzie mógł zarządzać użytkownikami.
* Zapewnienie obsługi zapytań do bazy danych.
* Zabezpieczenia w postaci systemu logowania użytkownika.
* Strona główna aplikacji w postaci tablicy ( poglądowy projekt załącznik nr.1).
* Instalator aplikacji dla systemów z rodzinny Windows.

1. **Wymagania niefunkcjonalne**

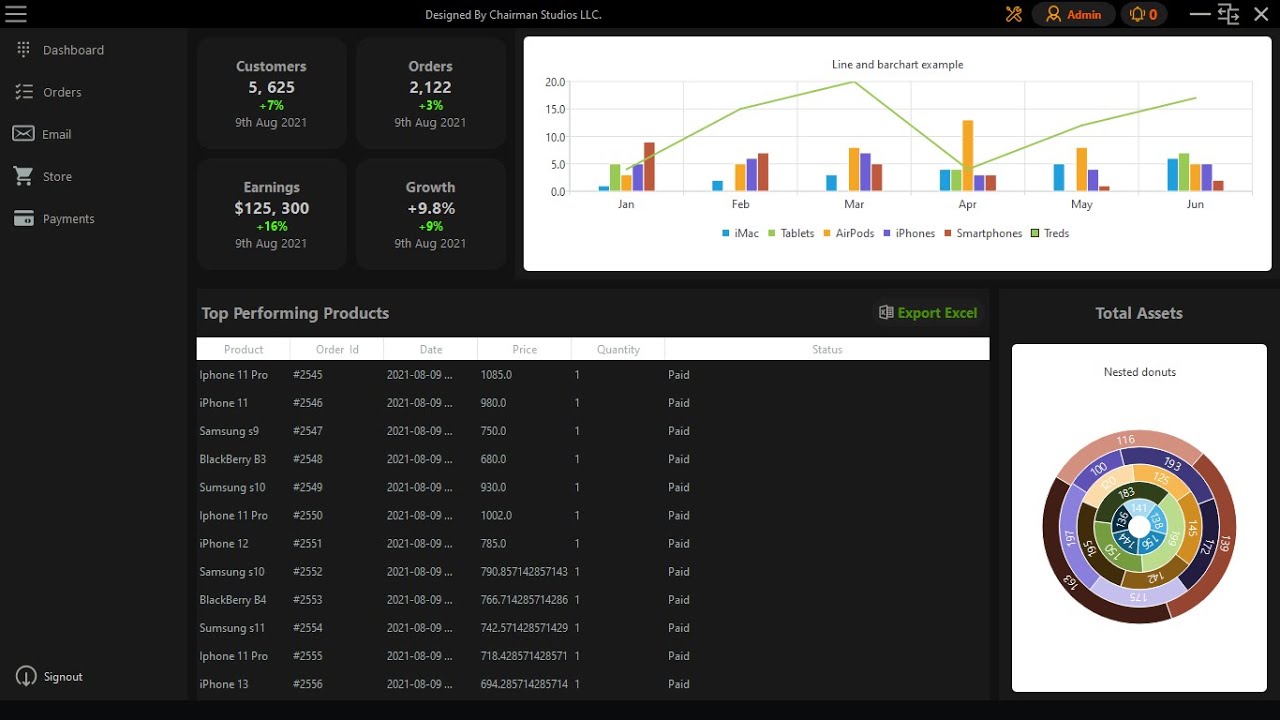
* Responsywny i intuicyjny interfejs użytkownika.
* Estetyczny i czytelny interfejs graficzny.
* Dane użytkowników oraz dane klientów powinny być przechowywane w bezpieczny sposób.
* Użytkownik powinien łatwo odnaleźć i korzystać z funkcji aplikacji.
* Użytkownik nie będzie mógł wprowadzać błędnych danych tj. tekst w miejsce pola numerycznego.
* Aplikacja powinna działać na platformie desktopowej Windows.
* Logowanie postępów prac na platformie Trello.
* Korzystanie z systemu kontroli wersji oprogramowania.
* Przeprowadzenie testów użytkowników końcowych przed oddaniem projektu.
* Wprowadzanie ulepszeń i naprawa zgłoszonych błędów.

1. **Wykorzystane Technologie**

* Język programowania Python
* Biblioteka GUI PyQ6
* Baza danych SQLite
* Biblioteki analityczne języka Python: NumPy, Pandas i Matplotlib
* System kontroli wersji: Platforma GitHub ( Załącznik nr.2)
* System Operacyjny Windows 11

1. **Przykładowa architektura bazy danych**
2. **Tabela "Accounts"**:
   * account\_id: unikalny identyfikator konta (klucz główny)
   * name: nazwa konta
   * address: adres konta
   * phone: numer telefonu konta
   * email: adres e-mail konta
3. **Tabela "Contacts"**:
   * contact\_id: unikalny identyfikator kontaktu (klucz główny)
   * account\_id: identyfikator przypisanego konta (klucz obcy z tabeli "Accounts")
   * first\_name: imię kontaktu
   * last\_name: nazwisko kontaktu
   * phone: numer telefonu kontaktu
   * email: adres e-mail kontaktu
4. **Tabela "Opportunity"**:
   * opportunity\_id: unikalny identyfikator możliwości (klucz główny)
   * account\_id: identyfikator powiązanego konta (klucz obcy z tabeli "Accounts")
   * amount: wartość możliwości
   * stage: etap możliwości (np. "Negocjacje", "Zamknięte")
   * closing\_date: data zamknięcia możliwości
5. **Tabela "Quote"**:
   * quote\_id: unikalny identyfikator oferty (klucz główny)
   * opportunity\_id: identyfikator powiązanej możliwości (klucz obcy z tabeli "Opportunity")
   * amount: wartość oferty
   * valid\_until: data ważności oferty
6. **Tabela "Order"**:
   * order\_id: unikalny identyfikator zamówienia (klucz główny)
   * account\_id: identyfikator powiązanego konta (klucz obcy z tabeli "Accounts")
   * product\_id: identyfikator powiązanego produktu (klucz obcy z tabeli "Product")
   * quantity: ilość zamówionych produktów
   * order\_date: data zamówienia
7. **Tabela "Product"**:
   * product\_id: unikalny identyfikator produktu (klucz główny)
   * name: nazwa produktu
   * price: cena produktu
8. **Tabela "Users"**:
   * user\_id: unikalny identyfikator usera (klucz główny)
   * username: login usera
   * pass: hasło usera
   * role: rola użytkownika np. admin
9. **Product Backlog**
10. **Utworzenie interfejsu użytkownika:**
    * Stworzenie interfejsu użytkownika, który umożliwi użytkownikom nawigację i interakcję z różnymi obiektami CRM, takimi jak konta, kontakty, możliwości, oferty, zamówienia itp.
11. **Implementacja operacji CRUD:**
    * Umożliwienie dodawania, odczytywania, aktualizacji i usuwania danych dla obiektów CRM, takich jak konta, kontakty, możliwości, oferty, zamówienia, produkty.
12. **Integracja z bazą danych SQLite:**
    * Stworzenie modułu integracyjnego, który umożliwi komunikację z bazą danych SQLite w celu przechowywania i pobierania danych dotyczących obiektów CRM.
13. **Wyszukiwanie i filtrowanie danych:**
    * Dodanie funkcjonalności wyszukiwania i filtrowania danych, aby użytkownicy mogli łatwo odnajdywać konkretne informacje na podstawie określonych kryteriów.
14. **Załadowanie danych z pliku CSV:**
    * Implementacja mechanizmu importu danych z pliku CSV, który umożliwi użytkownikom ładowanie masowych danych do aplikacji CRM.
15. **Generowanie analiz i wykresów:**
    * Wykorzystanie bibliotek analitycznych języka Python, takich jak PyQ6, do tworzenia analiz i wykresów na podstawie posiadanych danych CRM. Na przykład: analiza sprzedaży, analiza konwersji, analiza klientów, analiza trendów rynkowych itp.
16. **Autoryzacja i uwierzytelnianie:**
    * Wprowadzenie mechanizmu autoryzacji i uwierzytelniania, który zapewni dostęp tylko uprawnionym użytkownikom do danych i funkcjonalności aplikacji.
17. **Powiadomienia i alarmy:**
    * Dodanie funkcjonalności powiadomień i alarmów, które informują użytkowników o ważnych terminach, zdarzeniach lub aktualizacjach dotyczących klientów, kontaktów lub innych obiektów CRM.
18. **Personalizacja interfejsu użytkownika:**
    * Dodanie funkcji personalizacji interfejsu użytkownika, takich jak dostosowanie układu, wybór preferowanych kolumn, dostosowanie motywu itp.
19. **Zabezpieczenia danych:**
    * Implementacja mechanizmów ochrony danych, takich jak zabezpieczenia przed nieautoryzowanym dostępem.
20. **Załączniki**

**Załącznik nr. 1***Poglądowy projekt strony głównej aplikacji w postaci tablicy.*



*Źródło:* [*https://www.youtube.com/watch?v=zkl4x3tQ-ws&ab\_channel=ChairmanStudios*](https://www.youtube.com/watch?v=zkl4x3tQ-ws&ab_channel=ChairmanStudios)

**Załącznik nr. 2***Link URL do Systemu kontroli wersji oprogramowania.*

[*https://github.com/BROM100/Aplikacja-CRM*](https://github.com/BROM100/Aplikacja-CRM)