

Dokumentacja projektu wykonywanego w ramach zajęć BAZY DANYCH I Poradnia specjalistyczna

Bartłomiej Długosz

I. Projekt koncepcji, założenia

1. Zdefiniowanie tematu projektu:

Poradnia specjalistyczna: psychologiczno-pedagogiczna umożliwia na umawianie pacjentów w poradni u różnych specjalistów, którzy dostępni są w danych placówkach.

Głównym założeniem projektu jest realizacja podstawowych zadań związanych z przetwarzaniem danych w bazie postgres.

Celem aplikacji jest korzystanie z istniejącej bazy danych, tabel oraz relacji związanymi między nimi. Istnieje możliwość wprowadzenia danych jak i ich odczyt.

2. Analiza wymagań użytkownika:

Użytkownik ma możliwość podglądu na całą bazę danych, wyboru placówki, w której chce podjąć działania takie jak: rejestracja pacjenta, umówienie wizyty, dodanie recepcjonisty, dodanie specjalisty, dodanie gabinetu, a także usunięcie niektórych danych z bazy.

3. Zaprojektowanie funkcji:

Z głównego menu:

wybierz placówkę – pozwala na wybór placówki, w której użytkownik chce wprowadzać dane / mieć wgląd na dane istniejące w tej placówce,

zamknij aplikację – kończy pracę z aplikacją,

Z menu placówki:

odśwież – aktualizuje menu o nowe dane,

zarejestruj pacjenta - pozwala na zarejestrowanie pacjenta w placówce,

dodaj recepcjonistę – pozwala na dodanie recepcjonisty w danej placówce,

dodaj gabinet – pozwala na dodanie w placówce nowego gabinetu, w którym mogą przyjmować specjaliści,

umów wizytę – pozwala na umówienie wizyty dla pacjentów w danej placówce,

dodaj specjalistę – pozwala na dodanie specjalisty w danej placówce,
powrót – powrót do menu głównego (wyboru placówki)s

II. Projekt diagramów (konceptualny)

4. Zdefiniowanie encji (obiektów) oraz ich atrybutów.

Wizyty – id, nazwa_wizyty, cena,

Wizyty_Pacjenci – id_wizyty, id_pacjenta, id_specjalisty, data_wizyty,

Pacjenci – id, imię, nazwisko, wiek, nr_telefonu, id_placowki,

Pacjenci_Dolegliwosci – id_pacjenta, id_dolegliwosci,

Dolegliwosci – id, nazwa,

Gabinety – id, numer_gabinetu

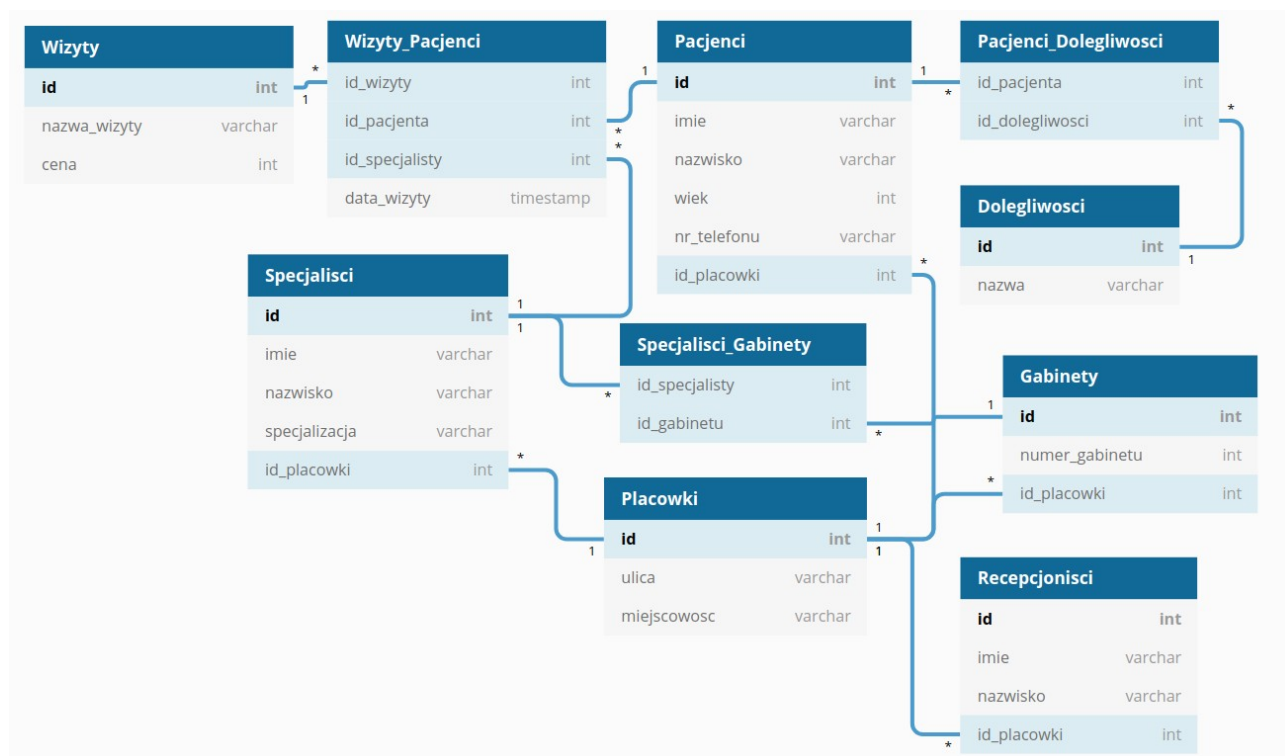
Recepcjonisci – id, imię, nazwisko, id_placowki,

Placowki – id, ulica, miejscowosc,

Specjalisci – id, imię, nazwisko, id_specjalnosc, specjalizacja, id_placowki,

Specjalisci_Gabinety – id_specjalisty, id_gabinetu,

5. Zaprojektowanie relacji pomiędzy encjami:



źródło : <https://dbdiagram.io/>

III. Projekt logiczny

6. Projektowanie tabel, kluczy, indeksów:

Tabela Wizyty:

id – identyfikuje jednoznacznie każdy rodzaj wizyty, połączenie z tabelą **Wizyty_Pacjenci** (1:N)

Tabela Wizyty_Pacjenci:

id_wizyty – jest to klucz obcy tabeli **Wizyty** (N:1),

id_pacjenta – jest to klucz obcy tabeli **Pacjenci** (N:1),

id_specjalisty – jest to klucz obcy tabeli **Specjalisci** (N:1)

Tabela Pacjenci:

id – identyfikuje jednoznacznie każdego pacjenta, połącznie z tabelami:

Wizyty_Pacjenci (1:N), **Pacjenci_Dolegliwosci** (1:N),

id_placowki – jest to klucz obcy tabeli **Placowki** (N:1)

Tabela Pacjenci_Dolegliwosci:

id_pacjenta – jest to klucz obcy tabeli **Pacjenci** (N:1),

id_dolegliwosci – jest to klucz obcy tabeli **Dolegliwosci** (N:1)

Tabela Dolegliwosci:

id – identyfikuje jednoznacznie każdy rodzaj dolegliwości, połączenie z tabelą **Pacjenci_Dolegliwosci** (1:N)

Tabela Gabinety:

id – identyfikuje jednoznacznie każdy gabinet, połączenie z tabelą

Specjalisci_Gabinety (1:N),

id_placowki - jest to klucz obcy tabeli **Placowki** (N:1)

Tabela Recepcjonisci:

id – identyfikuje jednoznacznie każdego recepcjonistę,

id_placowki - jest to klucz obcy tabeli **Placowki** (N:1)

Tabela Placowki:

id – identyfikuje jednoznacznie każdą placówkę, połączenie z tabelami: **Specjalisci** (1:N), **Gabinety** (1:N), **Pacjenci** (1:N), **Recepcjonisci** (1:N),

Tabela Specjalisci:

id – identyfikuje jednoznacznie każdego specjalistę, połączenie z tabelami:

Specjalisci_Gabinety (1:N), **Wizyty_Pacjenci** (1:N),

id_placowki – jest to klucz obcy tabeli **Placowki** (N:1),

Tabela Specjalisci_Gabinety:

id_specjalisty – jest to klucz obcy tabeli **Specjalisci** (N:1),

id_gabinetu – jest to klucz obcy tabeli **Gabinety** (N:1)

7. Słowniki danych:**Tabela Wizyty:**

id – integer,

nazwa_wizyty – varchar,

cena – integer, > 0

Tabela Wizyty_Pacjenci:

id_wizyty – integer, > 0

id_pacjenta – integer, >0

id_specjalisty – integer, >0

data_wizyty - timestamp

Tabela Pacjenci:

id – integer >0

imie – varchar, długość >= 3

nazwisko – varchar, długość >= 3

wiek – integer, >0, <100,

nr_telefonu – varchar, długość = 9

id_placowki – integer > 0

Tabela Pacjenci_Dolegliwosci:

id_pacjenta – integer, > 0

id_dolegliwosci – integer, > 0

Tabela Dolegliwosci:

id – integer, > 0

nazwa – varchar,

Tabela Gabinety:

id – integer, >0,

numer_gabinetu – integer, >0

id_placowki – integer, >0

Tabela Recepcjonisci:

id – integer, >0

imie – varchar, długość >= 3

nazwisko – varchar, długość >= 3

id_placowki – integer, >0

Tabela Placowki:

id – integer, >0
ulica – varchar,
miestowosc – varchar,

Tabela Specjalisci:

id – integer, >0,
imie – varchar, długość >= 3
nazwisko – varchar, długość >= 3,
specjalizacja – varchar,
id_placowki – integer, >0

Tabela Specjalisci_Gabinety:

id_specjalisty – integer, >0,
id_gabinetu – integer, >0,

8. Analiza zależności funkcyjnych i normalizacja tabel.

Pierwsza postać normalna (1NF):

W każdej tabeli jest jednoznacznie definiujący encję klucz główny. Wszystkie dane są atomowe.

Druga postać normalna (2NF):

Każda tabela przechowuje dane dotyczące tylko konkretnej klasy obiektów. Każdy niekluczowy atrybut jest nieredukowalnie zależny od klucza głównego.

Trzecia postać normalna (3NF):

Wszystkie niekluczowe atrybuty (niewchodzące w skład klucza głównego) są wzajemnie niezależne oraz nieredukowalnie zależne od klucza głównego.

9.Zaprojektowanie operacji na danych:

Wszystkie kwerendy używane w aplikacji znajdują w plikach **.sql** znajdujących się w folderze **sql**.

IV. Projekt funkcjonalny

10. Interfejsy do prezentacji, edycji i obsługi danych:

Główne menu: wgląd do **wszystkich tabel**, wybranie placówki, w której istnieje możliwość wprowadzenia poniższych danych:

Formularz rejestracji pacjenta: imię, nazwisko, wiek, telefon,

Formularz dodania recepcjonisty: imię, nazwisko

Formularz dodania gabinetu: numer gabinetu

Formularz umówienia wizyty dla pacjenta: pacjent, specjalista, wizyta, data wizyty, dolegliwość

Formularz dodania specjalisty: imię, nazwisko, gabinet, specjalizacja

Formularze są ściśle powiązane z tabelami: zarejestrowani pacjenci mogą zostać umówieni na wizyty u dostępnych specjalistów, z dostępnych dolegliwości, specjaliści mogą przyjmować w dostępnych gabinetach. W każdej placówce przyjmują inni specjaliści, są zarejestrowani różni pacjenci, jest wiele gabinetów oraz obsługują różni recepcjoniści.

11. Wizualizacja danych:

Wgląd **na wszystkie dane** z poziomu głównego menu aplikacji poprzez tzw. **listbox'y** (widżety modułu Tkinter języka Python).

Komunikaty informujące o potwierdzeniu dodania rekordu do bazy, a także w przypadku błędu walidacji - stosowne wskazówki co do poprawności danych.

12. Zdefiniowanie panelu sterowania aplikacji

Główne menu:

- **przyciski:**

przejdź (w zależności od wybranej placówki, przechodzimy do odpowiedniej), **zamknij aplikację** (kończy pracę z programem)

- **listbox'y:**

wszystkie rekody wszystkich tabel istniejących w bazie danych. Lista placówek ma wpływ na przycisk przejdź.

Menu placówki:

- **przyciski:**

przejdź (w zależności od wybranej placówki, przechodzimy do odpowiedniej), **zamknij aplikację** (kończy pracę z programem), **odśwież** - aktualizuje menu o ewentualnie nowo dodane dane w bazie oraz gdy nastąpi usunięcie dowolnego rekordu.

- **listbox'y:**

wszystkie rekody wszystkich tabel istniejących w danej placówce. Możliwość usunięcia wybranych rekordów

Rejestracja pacjenta:

- **przyciski: odśwież, zarejestruj** – następuje walidacja wprowadzonych danych oraz rejestracja pacjenta w bazie, **powrót** - powrót do menu placówki,
- **formularz rejestracji**: imię, nazwisko, wiek oraz numer telefonu

Rejestracja recepcjonisty:

- **przyciski: odśwież, dodaj** – następuje walidacja wprowadzonych danych oraz dodanie recepcjonisty do bazy, **powrót** - powrót do menu placówki,
- **formularz rejestracji**: imię, nazwisko

Dodanie gabinetu:

- **przyciski: odśwież, dodaj** – następuje walidacja wprowadzonych danych oraz dodanie gabinetu do bazy, **powrót** - powrót do menu placówki,
- **formularz rejestracji**: numer gabinetu

Umawianie wizyty:

- **przyciski: odśwież, umów wizytę** – następuje walidacja wprowadzonych danych oraz umówienie wizyty dla pacjenta, **powrót** - powrót do menu placówki,
- **formularz rejestracji**: pacjent, specjalista, rodzaj wizyty, data wizyty, dolegliwość pacjenta

Dodanie specjalisty:

- **przyciski: odśwież, dodaj** – następuje walidacja wprowadzonych danych oraz dodanie specjalisty do bazy, **powrót** - powrót do menu placówki,
- **formularz rejestracji**: imię, nazwisko, gabinet, w którym będzie przyjmował specjalista, wybór specjalizacji dla specjalisty

13. Makropolecenia:

W menu umawiania wizyty, pacjenta można wybrać jedynie z listy zarejestrowanych pacjentów w danej placówce, podobnie jak i specjalistę, który będzie przyjmował pacjenta, a także istnieje możliwość wybrania rodzaju wizyty oraz dolegliwości pacjenta.

W menu dodania specjalisty do placówki istnieją dwie pomocne listy, w których wybrać można gabinet, w którym będzie przyjmował dany specjalista oraz można wybrać jego specjalność z definiowanej listy specjalności.

Dużym ułatwieniem jest początkowe wybranie placówki, w której dokonuje się większość operacji na bazie danych.

V. Dokumentacja

14. Wprowadzanie danych:

Wszystkie tabele zostały wstępnie uzupełnione przykładowymi danymi oraz możliwe jest wprowadzanie danych do większości tabel:

Dane wprowadzanie ręcznie:

Tabele:

Wizyty_Pacjenci, Pacjenci, Specjaliści, Gabinety, Recepcjonisci

Dane wprowadzone automatycznie:

Tabele:

Wizyty, Pacjenci_Dolegliwosci, Dolegliwosci, Placówki, Specjaliści_Gabinety

15. Dokumentacja użytkownika:

Użytkownik ma możliwość wglądu do wszystkich tabel z głównego menu. Wybranie odpowiedniej placówki z listy oraz kliknięcie przyciska 'Przejdź' przenosi w odpowiednie miejsce.

W placówce użytkownik ma możliwość rejestracji pacjentów, dodania: recepcjonistów, specjalistów oraz gabinetów, a także umawianie wizyt dla pacjentów zarejestrowanych w danej placówce. Ma wgląd na podgląd umówionych wizyt, a także na podgląd specjalistów, którzy przyjmują w danej placówce.

16. Opracowanie dokumentacji technicznej:

Kod został udokumentowany w kodzie źródłowym.

17. Wykaz literatury

- <https://docs.python.org/3/> ,
- <https://docs.python.org/3/library/tkinter.html> ,
- <https://stackoverflow.com/> ,
- <https://dashboard.heroku.com/> ,
- <https://dbdiagram.io/>