



AGH

Projekt z przedmiotu Bazy Danych I

Wizualizacja algorytmów sortowania

Dokumentacja

Daria Kokot

Styczeń 2024

Dokumentacja projektu wykonywanego w ramach zajęć BAZY DANYCH I

I. Projekt koncepcji, założenia (3 pkt.)

1. Zdefiniowanie tematu projektu: krótki opis zadania, zdefiniowanie jego celów i zadań jakie ma realizować.

Celem projektu było stworzenie bazy danych reprezentującej system zarządzania projektami. Systemy takie często wykorzystywane są przez zespoły IT oraz biznesowe. Projekt implementuje funkcjonalność potrzebną, aby dane w bazie, dodawać, usuwać dane z bazy oraz tworzyć na ich podstawie raporty.

Aplikacja składa się z części bazodanowej (relacyjna baza PostgreSQL) oraz interfejsu użytkownika (backend: FastApi, frontend: HTML, JS, CSS).

2. Analiza wymagań użytkownika: określenie funkcjonalności jakie ma spełniać projektowana baza danych.

Baza danych ma reprezentować:

- zbiór projektów danej firmy,
- tablice, z których składają się projekty; reprezentują część projektu, w którym znajduje się zadanie
- zadania należące do poszczególnych tablic; są to czynności, które należy wykonać w ramach projektu
- pracowników firmy, ich stanowiska oraz role w poszczególnych projektach,
- relacje pomiędzy poszczególnymi zadaniami i typy tych relacji.

Dane można dodawać do każdej z tabel za pomocą interfejsu użytkownika. Na podstawie danych można tworzyć raporty.

Raporty przedstawiają:

- tablice bez zadań,
- tablice z zadaniami,
- tablicę z sumą punktów z zadań mniejszą niż 100,
- tablicę z sumą punktów z zadań większą niż 100,
- informację o liczbie tablic w projekcie oraz liczby zadań w projekcie,
- projekty przypisane do danego użytkownika.

Podczas dodawania nowego użytkownika po stronie bazy powinien być weryfikowany numer telefonu spełniający następujące warunki: opcjonalny numer kierunkowy poprzedzony plusem, 7-15 znaków, brak spacji.

Podczas dodawania nowego użytkownika po stronie bazy powinien być weryfikowany adres e-mail użytkownika. Tak, aby był zgodny z ogólnie przyjętymi standardami.

Użytkownik loguje się do systemu za pomocą adresu e-mail i hasła.

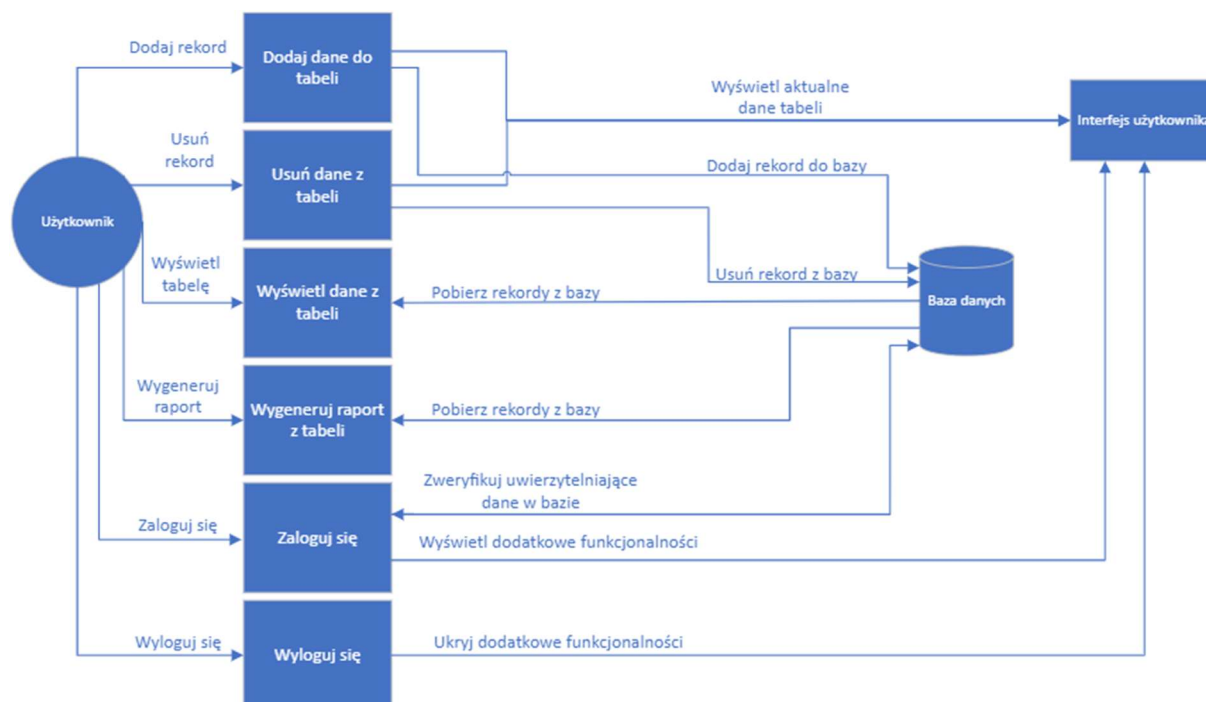
Każdy administrator może dodawać uprawnienia innym użytkownikom.

3. Zaprojektowanie funkcji: określenie podstawowych funkcji realizowanych w bazie danych.

Funkcje wykonywane na bazie danych to wybieranie danych (SELECT), wstawianie (INSERT) oraz usuwanie (DELETE). Baza zawiera widoki zgodne z wymaganiami użytkownika

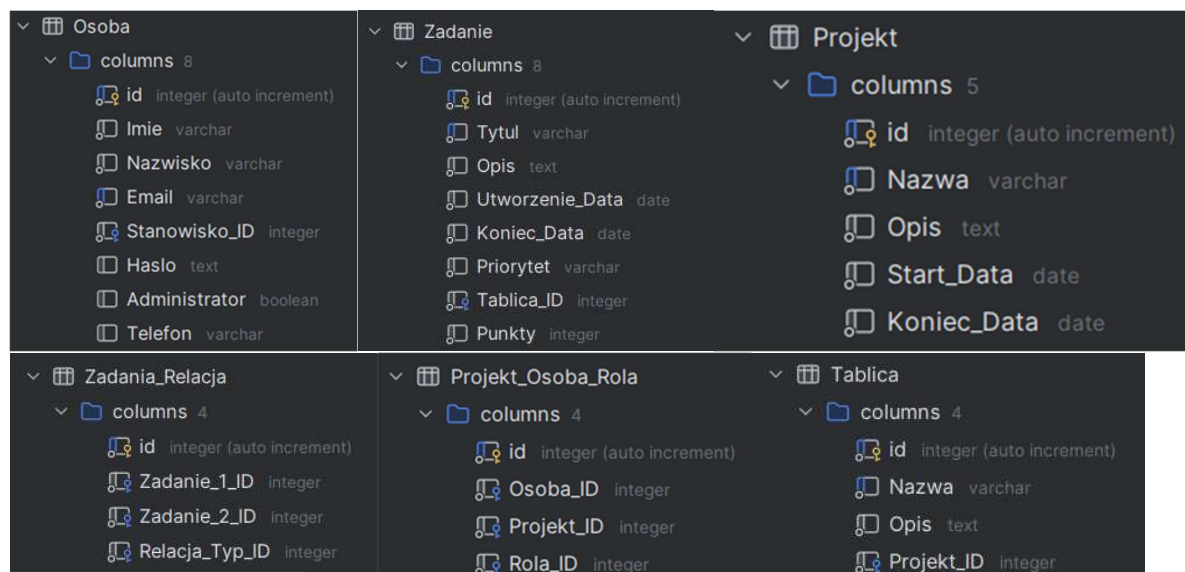
II. Projekt diagramów (konceptualny) (2 pkt.)

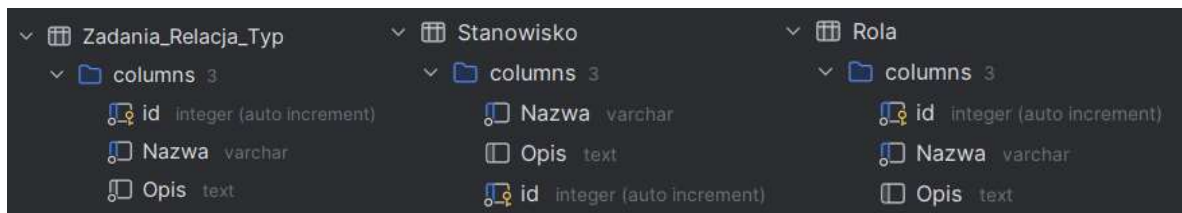
4. Budowa i analiza diagramu przepływu danych (DFD): określenie przepływu danych w systemie (wejscia, wyjścia, operacje, przechowywanie). Należy zastanowić się nad wybraniem elementów, które będą sterowały tym przepływem.



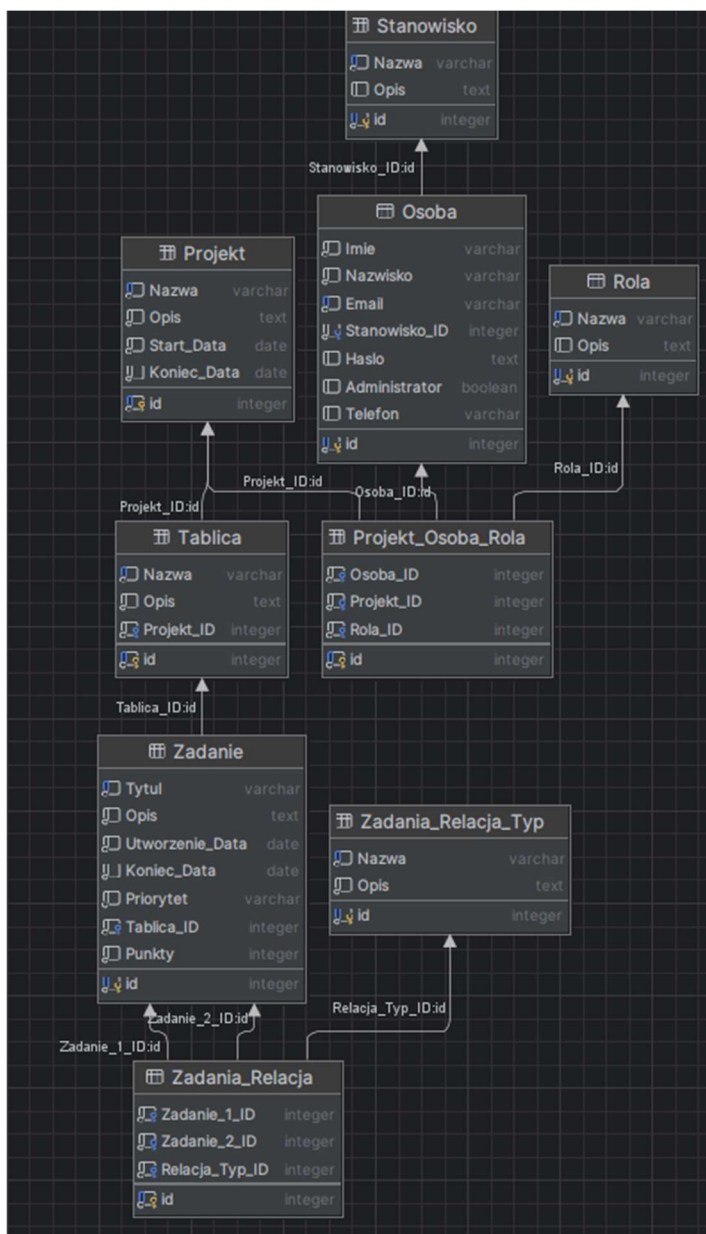
Rys. Diagram przepływu danych

5. Zdefiniowanie encji (obiektów) oraz ich atrybutów.





6. Zaprojektowanie relacji pomiędzy encjami: określenia klas encji wraz z atrybutami oraz zdefiniowanie kluczy. Następnie trzeba ustalić powiązania między encjami (**eliminacja powiązań wiele-do-wielu**). Etap ten powinien prowadzić do ostatecznej definicji logiki bazy danych. Efekt powinien zostać przedstawiony w postaci diagramu ERD.



Rys. Diagram ERD bazy danych (pełny diagram ERD w załączniku ERD_projekt.pdf)

III. Projekt logiczny (2 pkt.)

7. Projektowanie tabel, kluczy, indeksów: w oparciu o zdefiniowany diagram ERD należy opisać strukturę bazy danych wraz ze szczegółami technicznymi. Projekt w języku SQL znajduje się w załączniku SQL_projekt.sql.

Tabela "Stanowisko":

- **Kolumny:**
 - **id** (klucz główny)
 - **Nazwa** (unikalna wartość)
 - **Opis**
- **Klucz główny: id**
- **Unikalna wartość: Nazwa**

Tabela "Rola":

- **Kolumny:**
 - **id** (klucz główny)
 - **Nazwa** (unikalna wartość)
 - **Opis**
- **Klucz główny: id**
- **Unikalna wartość: Nazwa**

Tabela "Projekt":

- **Kolumny:**
 - **id** (klucz główny)
 - **Nazwa** (unikalna wartość, minimalna długość 3 znaków)
 - **Opis**
 - **Start_Data**
 - **Koniec_Data**
- **Klucz główny: id**
- **Unikalna wartość: Nazwa**
- **Warunek:** Minimalna długość nazwy projektu to 3 znaki

Tabela "Osoba":

- **Kolumny:**
 - **id** (klucz główny)
 - **Imie**

- **Nazwisko**
- **Email** (unikalna wartość, sprawdzająca format adresu email)
- **Stanowisko_ID** (klucz obcy odnoszący się do tabeli "Stanowisko")
- **Haslo**
- **Administrator**
- **Telefon**
- **Klucz główny: id**
- **Klucz obcy: Stanowisko_ID** (referencja do tabeli "Stanowisko")
- **Unikalna wartość: Email**
- **Warunek:** Format poprawnego adresu email

Tabela "Tablica":

- **Kolumny:**
 - **id** (klucz główny)
 - **Nazwa** (unikalna wartość, minimalna długość 3 znaki)
 - **Opis**
 - **Projekt_ID** (klucz obcy odnoszący się do tabeli "Projekt")
- **Klucz główny: id**
- **Klucz obcy: Projekt_ID** (referencja do tabeli "Projekt")
- **Unikalna wartość: Nazwa**
- **Warunek:** Minimalna długość nazwy tablicy to 3 znaki

Tabela "Zadanie":

- **Kolumny:**
 - **id** (klucz główny)
 - **Tytuł** (unikalna wartość)
 - **Opis**
 - **Utworzenie_Data**
 - **Koniec_Data**
 - **Priorytet**
 - **Tablica_ID** (klucz obcy odnoszący się do tabeli "Tablica")
 - **Punkty**
- **Klucz główny: id**

- **Klucz obcy:** **Tablica_ID** (referencja do tabeli "Tablica")
- **Unikalna wartość:** **Tytul**

Tabela "Zadania_Relacja_Typ":

- **Kolumny:**
 - **id** (klucz główny)
 - **Nazwa** (unikalna wartość)
 - **Opis**
- **Klucz główny:** **id**
- **Unikalna wartość:** **Nazwa**

Tabela "Zadania_Relacja":

- **Kolumny:**
 - **id** (klucz główny)
 - **Zadanie_1_ID** (klucz obcy odnoszący się do tabeli "Zadanie")
 - **Zadanie_2_ID** (klucz obcy odnoszący się do tabeli "Zadanie")
 - **Relacja_Typ_ID** (klucz obcy odnoszący się do tabeli "Zadania_Relacja_Typ")
- **Klucz główny:** **id**
- **Klucze obce:**
 - **Zadanie_1_ID** (referencja do tabeli "Zadanie")
 - **Zadanie_2_ID** (referencja do tabeli "Zadanie")
 - **Relacja_Typ_ID** (referencja do tabeli "Zadania_Relacja_Typ")

Tabela "Projekt_Osoba_Rola":

- **Kolumny:**
 - **id** (klucz główny)
 - **Osoba_ID** (klucz obcy odnoszący się do tabeli "Osoba")
 - **Projekt_ID** (klucz obcy odnoszący się do tabeli "Projekt")
 - **Rola_ID** (klucz obcy odnoszący się do tabeli "Rola")
- **Klucz główny:** **id**
- **Klucze obce:**
 - **Osoba_ID** (referencja do tabeli "Osoba")
 - **Projekt_ID** (referencja do tabeli "Projekt")
 - **Rola_ID** (referencja do tabeli "Rola")

8. Słowniki danych: **wypisać słownik danych, określić dziedziny oraz ich ograniczenia.**

Tabela "Zadania_Relacja_Typ":

- **Dziedzina:**
 - **id:** Numer identyfikacyjny typu relacji zadania.
 - **Nazwa:** Nazwa typu relacji zadania.
 - **Opis:** Opis typu relacji zadania.
- **Ograniczenia:**
 - **Nazwa:** Unikalna wartość. Każdy typ relacji zadania ma unikalną nazwę.

Słownik relacji:

Odnosi się do:

id: Numer identyfikacyjny: 8

Nazwa: Odnosi się do

Opis: Opis relacji "odnosi się do"

Duplikuje:

id: Numer identyfikacyjny: 9

Nazwa: Duplikuje

Opis: Opis relacji "duplikuje"

Jest duplikowane przez:

id: Numer identyfikacyjny: 10

Nazwa: Jest duplikowane przez

Opis: Opis relacji "jest duplikowane przez"

Blokuje:

id: Numer identyfikacyjny: 11

Nazwa: Blokuje

Opis: Opis relacji "blokuje"

Jest blokowane przez:

id: Numer identyfikacyjny: 12

Nazwa: Jest blokowane przez

Opis: Opis relacji "jest blokowane przez"

Klonuje:

id: Numer identyfikacyjny: 13

Nazwa: Klonuje

Opis: Opis relacji "klonuje"

Jest klonowane przez:

id: Numer identyfikacyjny: 18

Nazwa: Jest klonowane przez

Opis: Opis relacji "jest klonowane przez"

9. Analiza zależności funkcyjnych i normalizacja tabel (dekompozycja do 3NF ewentualnie BCNF):
dokonanie sprawdzenia, czy tabele spełniają warunki zakładanych postaci normalnych (ew.
dekompozycja w celu normalizacji).

Dokonano analizy zależności funkcyjnych:

1. Przejrzano definicje tabel i zidentyfikowano klucze główne i obce w każdej tabeli.
2. Zidentyfikowano relacje między tabelami (połączenia tabel za pomocą kluczy obcych)
3. Przeanalizowano dane w kolumnach, zidentyfikowano unikalne wartości, które mogły posłużyć jako potencjalne klucze kandydujące (potencjalne klucze główne)
4. Dokonano identyfikacji zależności funkcyjnych (wzajemne zależności kolumn w jednej tabeli, redundantne kolumny wydzielono do innych tabel)
5. Sprawdzono unikalność i relacje między kolumnami. Upewniono się, że każda kolumna ma unikalne wartości, szczególnie te, które są kluczami głównymi
6. Wykorzystano klauzule CONSTRAINT do definiowania unikalnych wartości lub innych zależności między kolumnami

Zdekomponowano tabele do 3NF stosując następujące zadasy:

Pierwsza postać normalna (1NF)

Relacja jest w pierwszej postaci normalnej, jeśli:

- opisuje jeden obiekt,
- wartości atrybutów są elementarne (atomowe, niepodzielne) – każda kolumna jest wartością skalarną (atomową), a nie macierzą lub listą czy też czymkolwiek, co posiada własną strukturę^[1],
- nie zawiera kolekcji (powtarzających się grup informacji)
- kolejność wierszy może być dowolna (znaczenie danych nie zależy od kolejności wierszy).

Druga postać normalna (2NF)

Relacja jest w drugiej postaci normalnej wtedy i tylko wtedy, gdy jest w I postaci normalnej i żadna kolumna niekluczowa nie jest częściowo funkcyjnie zależna od jakiegokolwiek klucza potencjalnego.

Trzecia postać normalna (3NF)

Relacja jest w trzeciej postaci normalnej wtedy i tylko wtedy, gdy jest w II postaci normalnej i żaden atrybut niekluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych.

10. Denormalizacja struktury tabel: ma ona na celu optymalizację przetwarzania, przechowywania danych archiwalnych, dostosowanie do specyficznych wymagań. [opcjonalnie].

Tego nie zrobiono.

11. Zaprojektowanie operacji na danych: zdefiniowanie kwerend dla realizacji funkcji wyspecyfikowanych w założeniach projektu (pierwsza część dokumentacji, punkt 3). Projekt w języku SQL znajduje się w załączniku SQL_projekt.sql.

W aplikacji wykorzystane zostały następujące typy kwerend:

Przeglądanie danych:

```
SELECT o.id, o."Imie", "Nazwisko", "Email", "Telefon", s."Nazwa"
FROM "Osoba" o
      INNER JOIN "Stanowisko" s ON s.id = o."Stanowisko_ID";
```

Dodawanie danych:

```
INSERT INTO "Osoba" ("Imie", "Nazwisko", "Email", "Telefon",
"Stanowisko_ID")
VALUES ('Jan', 'Kowalski', 'jan.kowalski@domena.pl', '+48 123 456 789', 1);
```

Usuwanie danych:

```
DELETE FROM "Osoba" WHERE "id" = 1;
```

W pliku załącznik1.sql znajduje się projekt w języku SQL.

*Powyżej przedstawiono tylko pojedyncze przypadki kwerend. Dla pozostałych tabeli kwerendy są analogiczne.

IV. Projekt funkcjonalny (2 pkt.)

12. Interfejsy do prezentacji, edycji i obsługi danych: zdefiniowanie struktury poszczególnych formularzy do wprowadzania danych oraz powiązań między formularzami.

Do każdej tabeli można wprowadzać dane za pomocą formularzy. Struktura poszczególnych formularzy oraz powiązania między nimi zostały zdefiniowane w kodzie aplikacji webowej.

13. Wizualizacja danych: określenie formy i struktury raportów które będą generowane przez bazę danych.

Raporty generowane są w postaci tabel oraz listy nazw. Są to raporty czasu rzeczywistego

Projekty

Moje projekty

Tablice

Zadania

Stanowiska

Osoby

Rola

Projekt_osoba_role

Zadania_relacja_typy

Zadania_relacje

Logowanie

Rejestracja

Logout

Lista projektów

Projekt	Nazwa	Projekt	Opis	Data rozpoczęcia	Data zakończenia	Imię	Nazwisko	Email	Rola	Nazwa	Rola	Opis	Akcje
Atlas		Aplikacja do obsługi map		2023-01-01	2023-12-31	B	B	bb@bb.com	a	v			<button>Usuń</button>
Atlas		Aplikacja do obsługi map		2023-01-01	2023-12-31	B	B	bb@bb.com	Developer	Pisze kod			<button>Usuń</button>
aaac		aaa		2024-01-04	2024-02-09	B	B	bb@bb.com	Developer	Pisze kod			<button>Usuń</button>

Rys. Raport przedstawiający projekty zalogowanego użytkownika jako tabela



Rys. Raport zawierający tablice z określoną liczbą zadań oraz określoną liczbą punktów jako lista nazw tabel

14. Zdefiniowanie panelu sterowania aplikacji.

Panel sterowania aplikacji to menu nawigacyjne widoczne na górze interfejsu użytkownika. Panel zmienia się w zależności, czy użytkownik jest zalogowany, czy nie. Na górze wyświetlany jest e-mail użytkownika. Pod menu znajduje się przycisk wylogowania.



Rys. Panel sterowania dla zalogowanego użytkownika



Rys. Panel sterowania dla zalogowanego użytkownika

15. Makropolecenia: powinny ułatwiać obsługę aplikacji z poziomu panelu sterowania oraz ewentualnie z poziomu formularzy.

W kodzie źródłowym można zidentyfikować zarówno makrozadania, jak i mikrozadania, które są elementami obsługi operacji asynchronicznych. Poniżej przedstawiam analizę w kontekście kodu (przykładowo z pliku `tablice.html`):

Makrozadania:

1. Funkcja **makeRequest** jest przykładem makrozadania. Wykorzystuje obiekt `XMLHttpRequest` do wykonywania operacji asynchronicznych, a jej wykonanie następuje w odpowiedzi na zdarzenia, takie jak zmiana stanu (**readystatechange**).
2. Funkcja **fetchData** również jest makrozadaniem. Wykorzystuje `XMLHttpRequest` do pobrania projektów i uruchamia się w odpowiedzi na zdarzenia związane z tym obiektem.

Mikrozadania:

1. Blok kodu wewnątrz **Promise.all(requestPromises).then(...)** reprezentuje mikrozadanie. Metoda **then** na obiektach `Promise` zawsze tworzy mikrozadanie, które zostanie wykonane po zakończeniu bieżącego makrozadania. W tym przypadku, obsługuje odpowiedzi wszystkich równoczesnych żądań asynchronicznych.

V. Dokumentacja (1 pkt.)

16. Wprowadzanie danych: zdefiniowanie sposobu wprowadzania danych (ręczne, automatyczne, import).

Dane wprowadzane są ręcznie za pomocą interfejsu użytkownika. Każdy zalogowany użytkownik może wprowadzać dane. Każdy administrator może dodawać uprawnienia innym użytkownikom.

17. Dokumentacja użytkownika: krótka instrukcja obsługi aplikacji.

Załącznik **Instrukcja_projekt.pdf**

18. Opracowanie dokumentacji technicznej: ma to być dokumentacja kodu, która pozwoli na dalszą rozbudowę systemu.

Dokumentacja kodu dostępna jest po otwarciu uruchomieniu aplikacji:

```
python -m uvicorn main:app --reload
```

Linku:

```
localhost:8000/docs
```

19. Wykaz literatury.

1. [https://pl.wikipedia.org/wiki/Posta%C4%87_normalna_\(bazy_danych\)](https://pl.wikipedia.org/wiki/Posta%C4%87_normalna_(bazy_danych))
2. <https://kursjs.pl/kurs/intervals/event-loop#mikrotask-makrotask>
3. SQL w 24 godziny. Wydanie VI (Ryan Stephens, Arie D. Jones, Ron Plew)
4. Database systems : a practical approach to design, implementation and management / Thomas M. Connolly, Carolyn E. Begg, [Anne Strachan].

20. Załączniki:

ERD_projekt.pdf

SQL_projekt.sql

Instrukcja_projekt.pdf

Zgodnosc_z_wymaganiami_projekt.pdf

requirements.txt

Folder ExportedData zawierający polecenia SQL z przykładowymi danymi systemu