

INŻYNIERIA OPROGRAMOWANIA  
II ROK

Dokumentacja projektu aplikacji  
**UBER EATS**

Autorzy:  
Norbert Armatys,  
Kacper Szczudło

## Spis treści

<b>Ogólna definicja zadań systemu.....</b>	<b>3</b>
<b>Listę bodźców zewnętrznych dostarczanych systemowi przez obiekty zewnętrzne.....</b>	<b>3</b>
<b>Podstawowe magazyny danych.....</b>	<b>3</b>
<b>Procesy przetwarzające.....</b>	<b>3</b>
<b>Diagram kontekstowy.....</b>	<b>4</b>
<b>Diagram przepływu danych – DFD.....</b>	<b>5</b>
Główne procesy.....	5
Poziom 1 – Logowanie.....	6
Poziom 2 – Zarządzanie kontami użytkowników.....	6
Poziom 3 – Zarządzanie restauracją.....	7
Poziom 4 - Składanie zamówień.....	7
Poziom 5 - Obsługa zamówień.....	8
Poziom 6 - Dodawanie komentarzy.....	8
<b>Słownik danych - DD.....</b>	<b>9</b>
<b>Diagram stanów - STD Diagram ogólny.....</b>	<b>10</b>
Diagram ogólny.....	10
Składanie zamówień.....	11
Obsługa zamówień.....	12
Zarządzanie restauracją.....	13
<b>Diagram związków encji - ERD.....</b>	<b>14</b>
<b>Specyfikacje procesów - PSPEC.....</b>	<b>15</b>

# Ogólna definicja zadań systemu

System Uber Eats został zaprojektowany do kompleksowej obsługi zamówień posiłków, składanych przez klientów, oraz ich dostawy. Obejmuje to zarządzanie kontami użytkowników, aktualizację menu, obsługę i weryfikację zamówień, a także dostawę i ocenę usług oraz produktów. Informacje o szczegółach zamówień są dostępne dla innych systemów, takich jak sprzedaż, baza danych restauracji oraz księgowość, co zapewnia płynne i zintegrowane działanie całego ekosystemu.

## Listę bodźców zewnętrznych dostarczanych systemowi przez obiekty zewnętrzne

- Klient
- Administrator
- Restauracja
- Dostawca

## Podstawowe magazyny danych

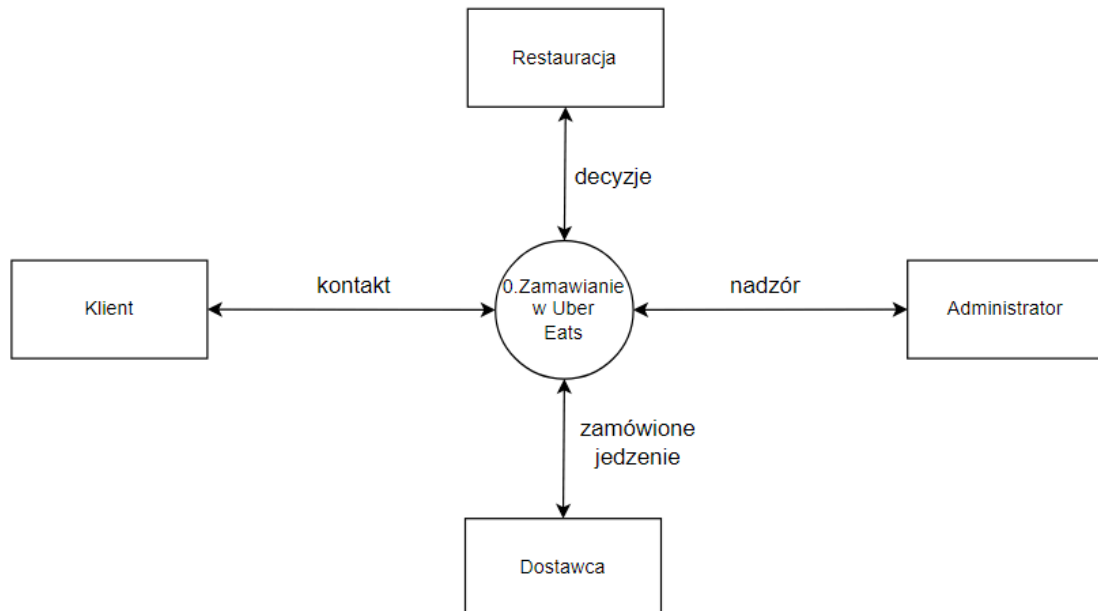
- Użytkownicy
- Restauracje
- Dostawcy
- Potrawy
- Zamówienia
- Koszyk
- Komentarze

## Procesy przetwarzające

- Logowanie
- Zarządzanie kontami użytkowników
- Zarządzanie restauracją
- Składanie zamówień
- Obsługa zamówień
- Dodawanie komentarzy

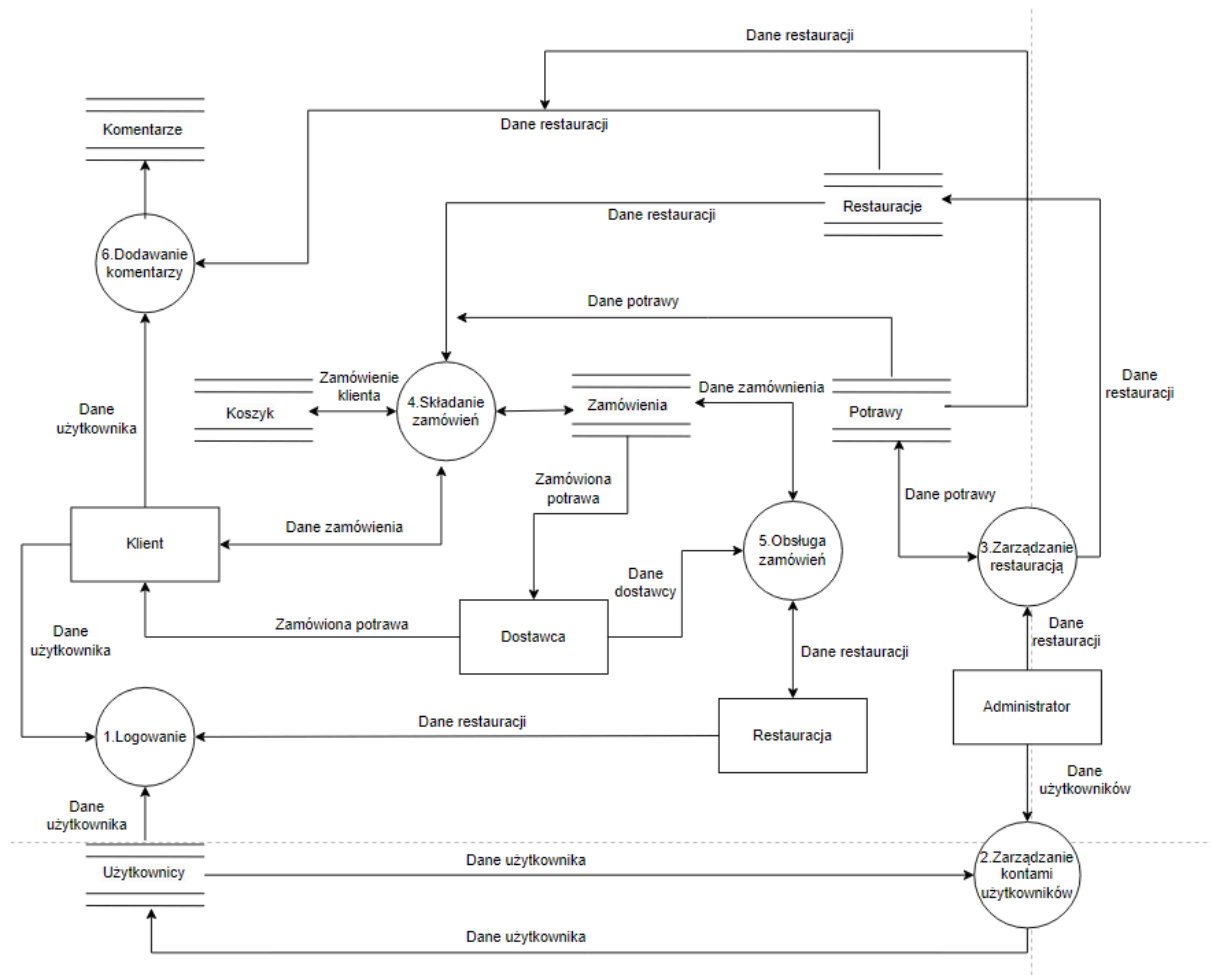
# Diagram kontekstowy

Diagram kontekstowy

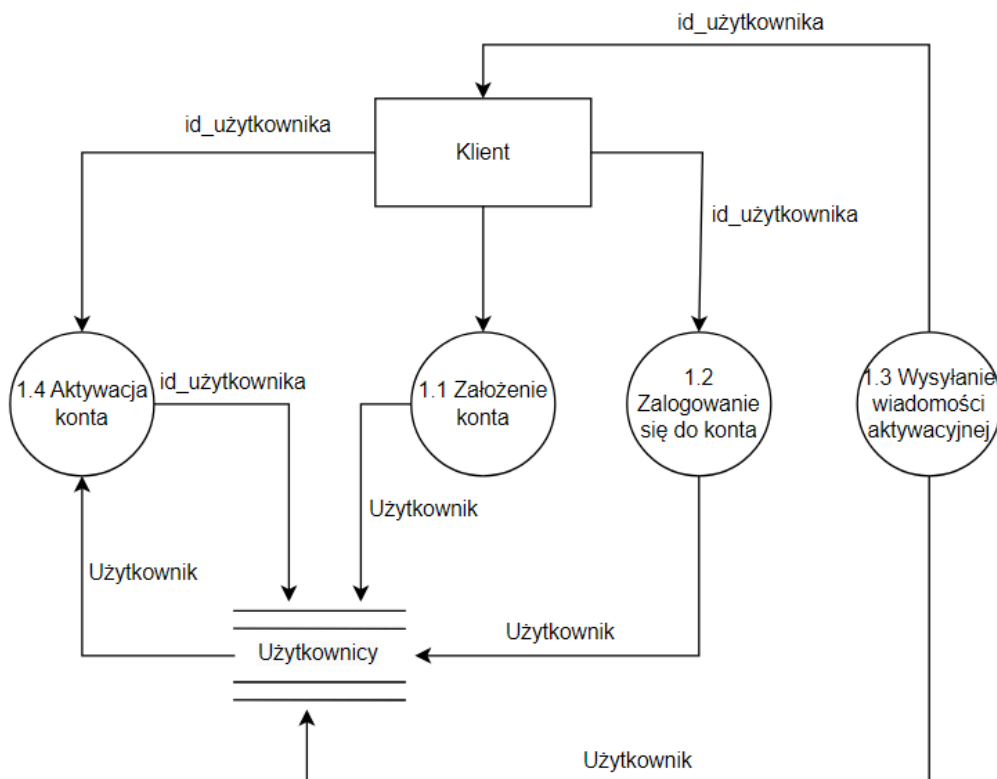


# Diagram przepływu danych – DFD

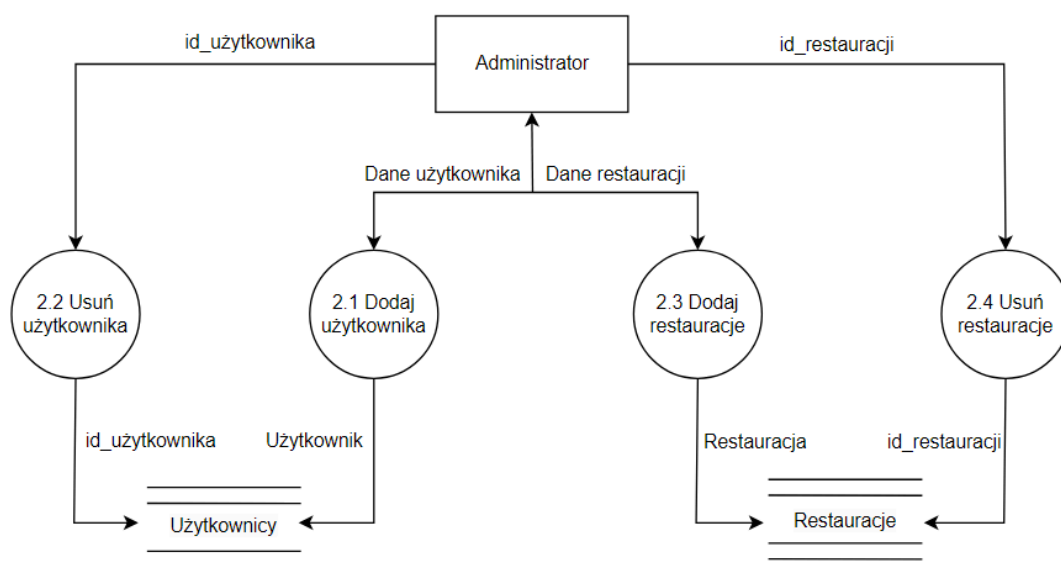
## Główne procesy



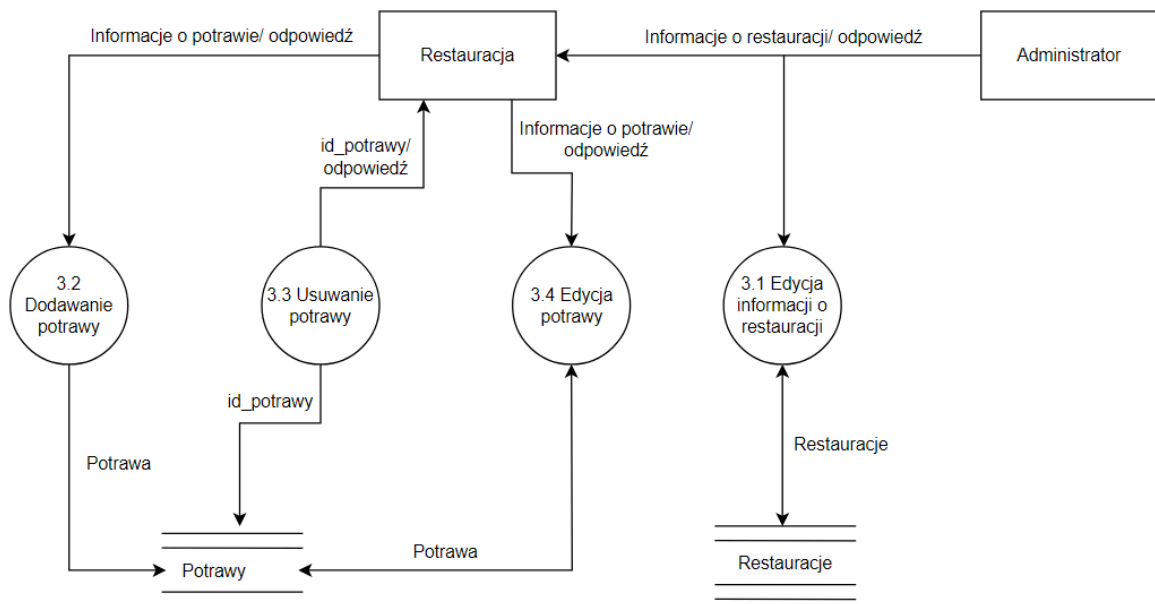
## Poziom 1 – Logowanie



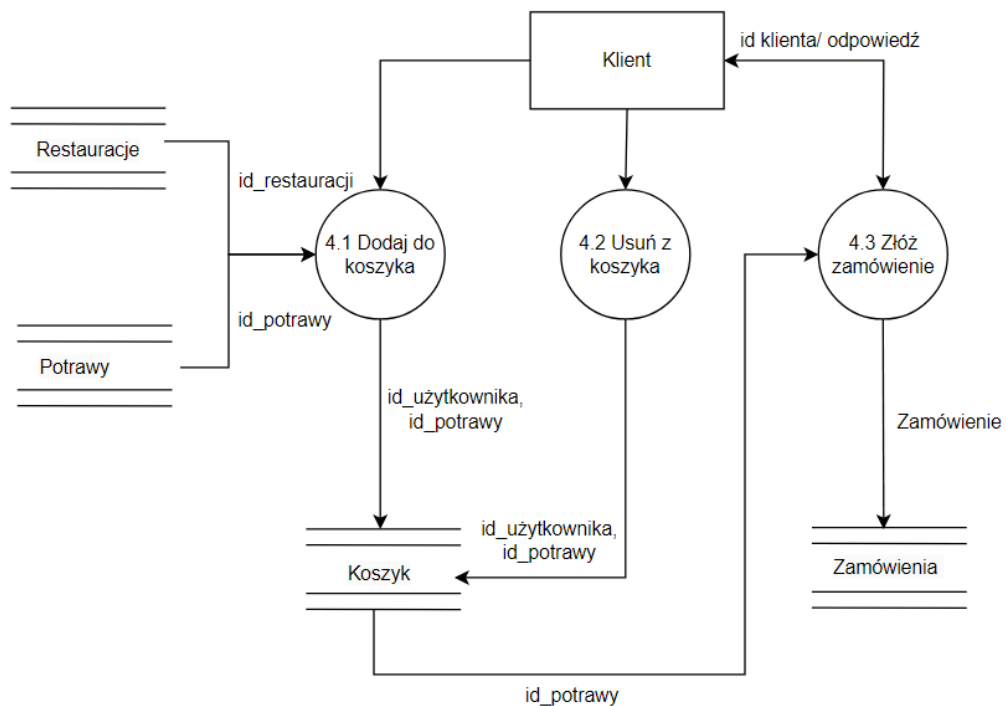
## Poziom 2 – Zarządzanie kontami użytkowników



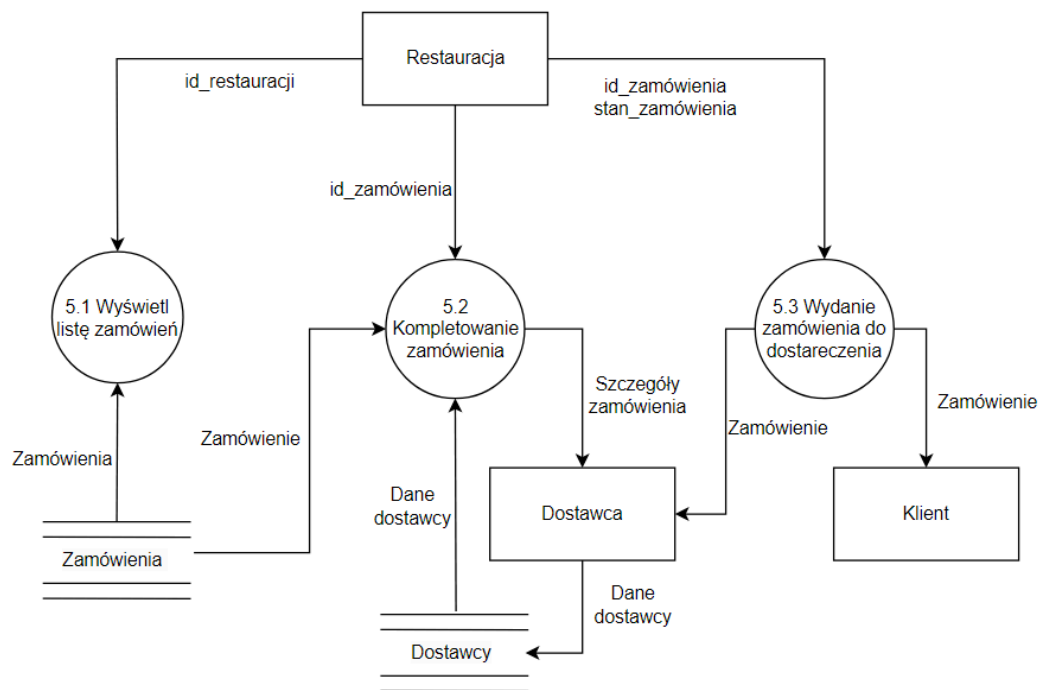
## Poziom 3 – Zarządzanie restauracją



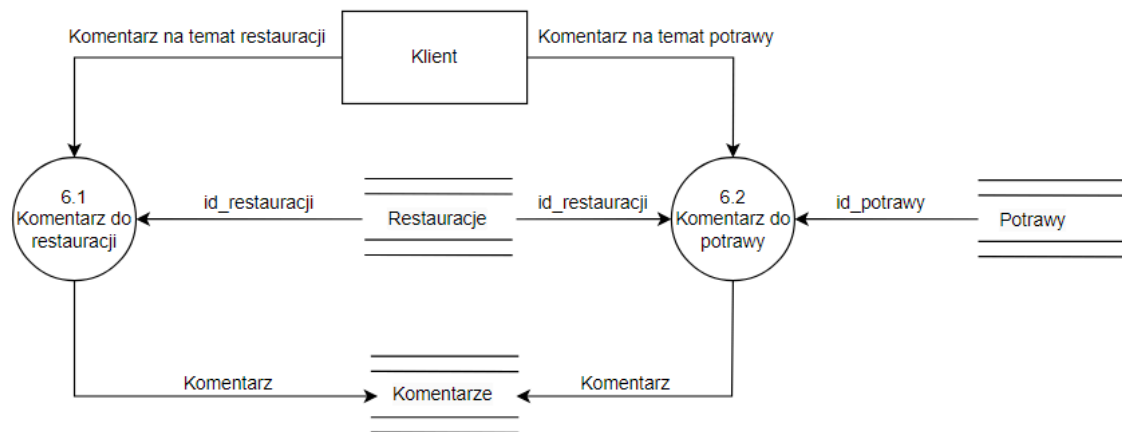
## Poziom 4 - Składanie zamówień



## Poziom 5 - Obsługa zamówień



## Poziom 6 - Dodawanie komentarzy





# Słownik danych - DD

Klient = ["Pan"|"Pani"] + Imię + (drugie Imię) + Nazwisko \*Również nazywany użytkownikiem\*

Dane użytkownika / klienta=

@Id\_użytkownika \*unikatowy numer identyfikacyjny dla poszczególnego klienta\*

Dane restauracji =

@id\_restauracji \*unikatowy numer identyfikacyjny dla poszczególniej restauracji\*

Dane potrawy =

@id\_potrawy \*unikatowy numer identyfikacyjny dla poszczególniej potrawy\*

Dane zamówienia =

@id\_zamówienia \*unikatowy numer identyfikacyjny dla poszczególnego zamówienia\*

@stan\_zamówienia \*Czy zamówienie jest przygotowywane, pakowane, transportowane\*

Informacje o potrawie /odpowiedź = dane o dostępności potrawy i zaakceptowanie bądź jej odrzucenie

Informacje o restauracji/ odpowiedź = dane o dostępności restauracji

Zamówienie klienta =

@id\_zamowienia \*unikatowy\* +  
id\_klienta + Data Zamówienia +  
Zamówiona Potrawa +  
(Charakterystyka Zamówienia

Kupuje =

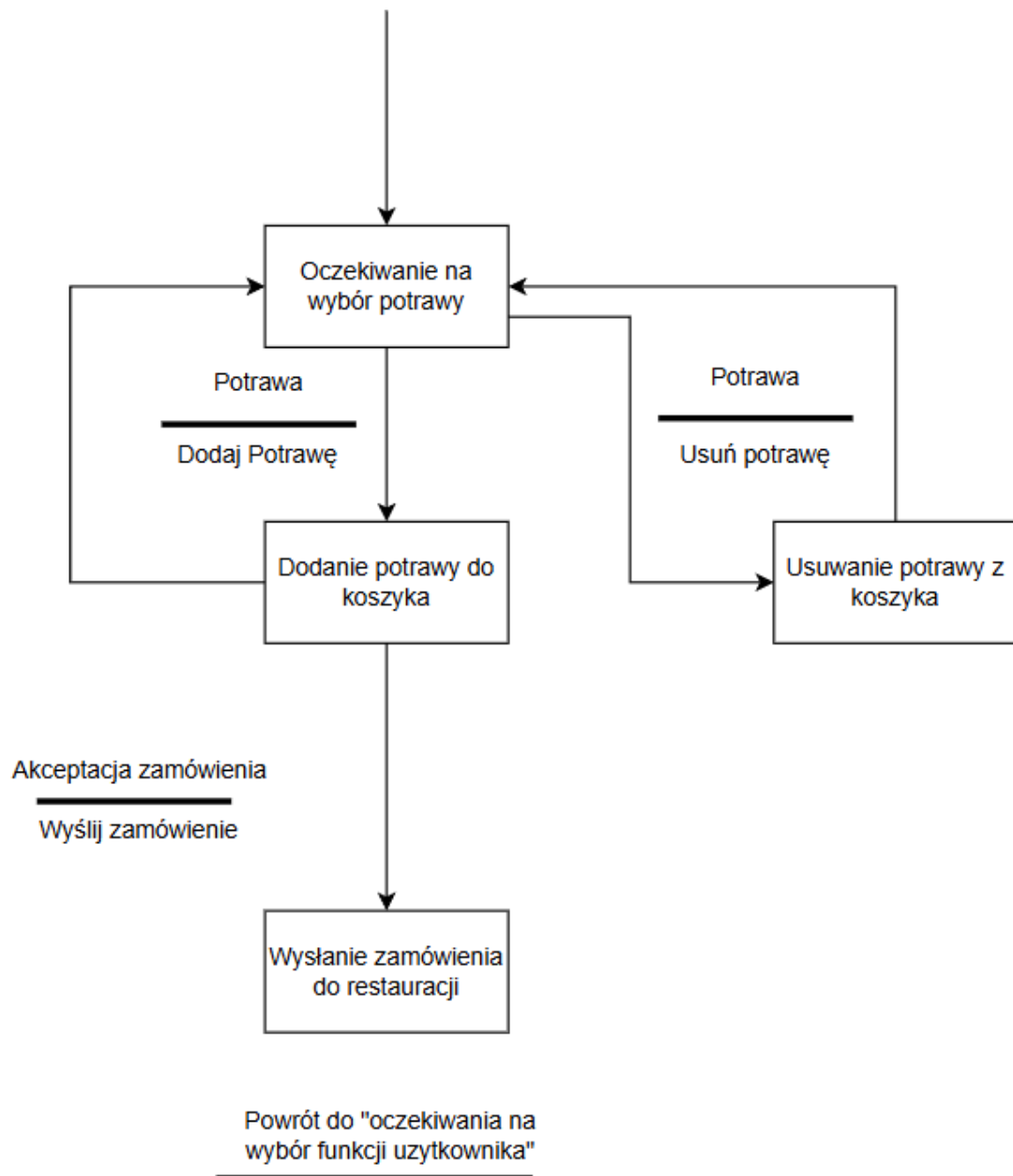
\*powiązanie klienta i jednego lub więcej towarów\*  
@ id\_klienta + 1{@id towaru + zakupiona ilość}

# Diagram stanów - STD Diagram ogólny

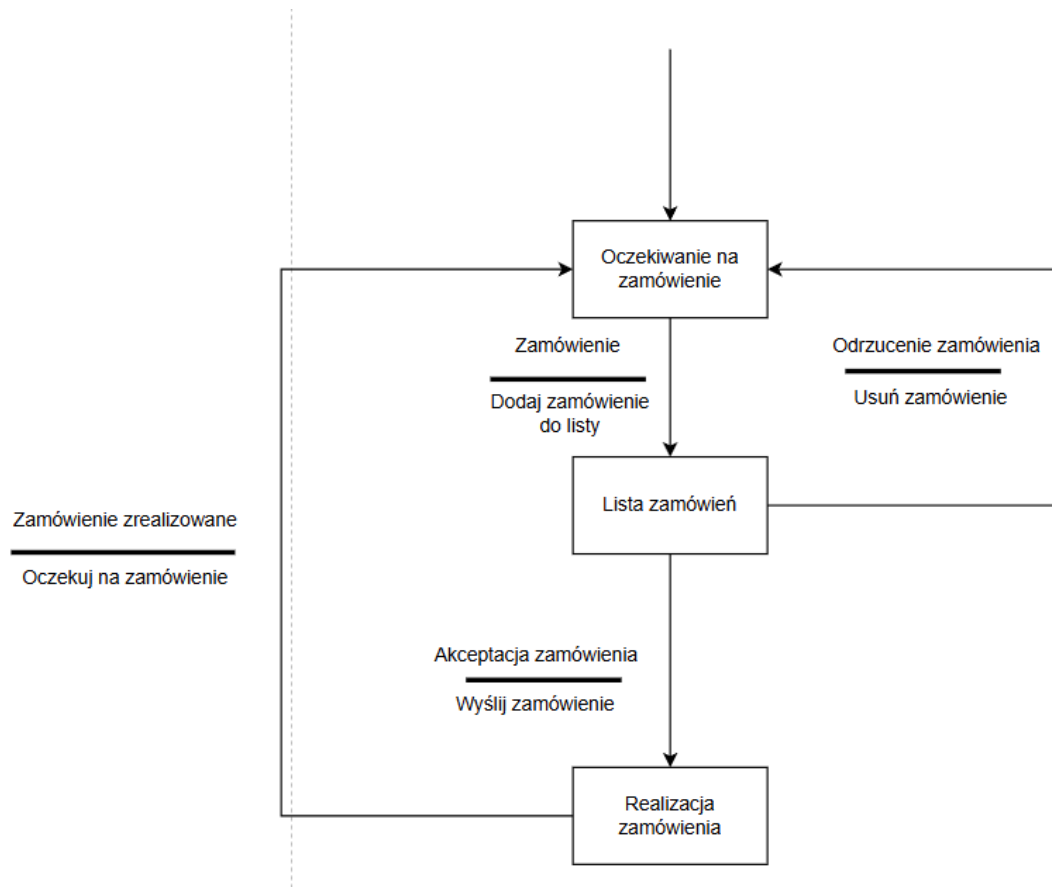
## Diagram ogólny



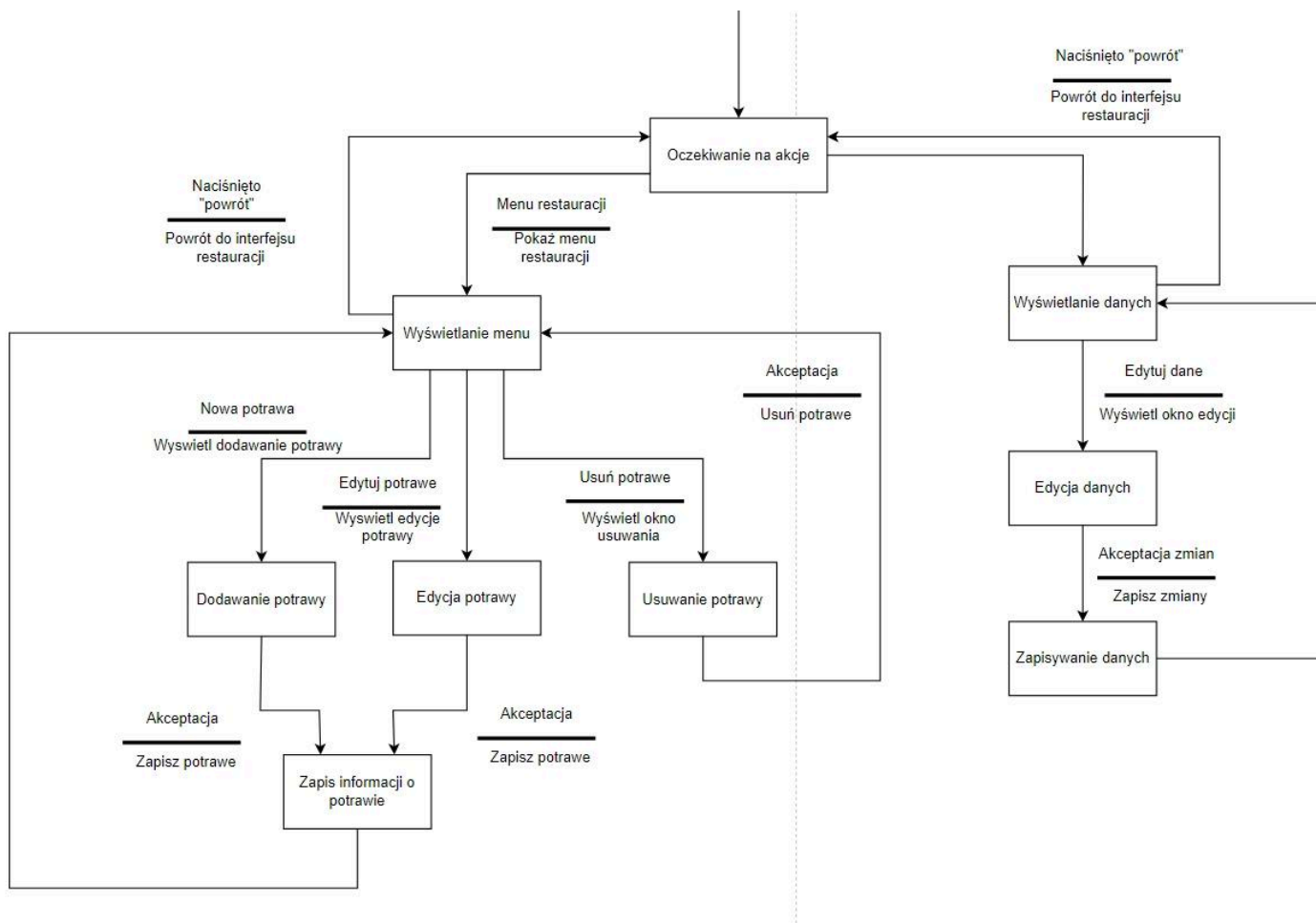
## Składanie zamówień



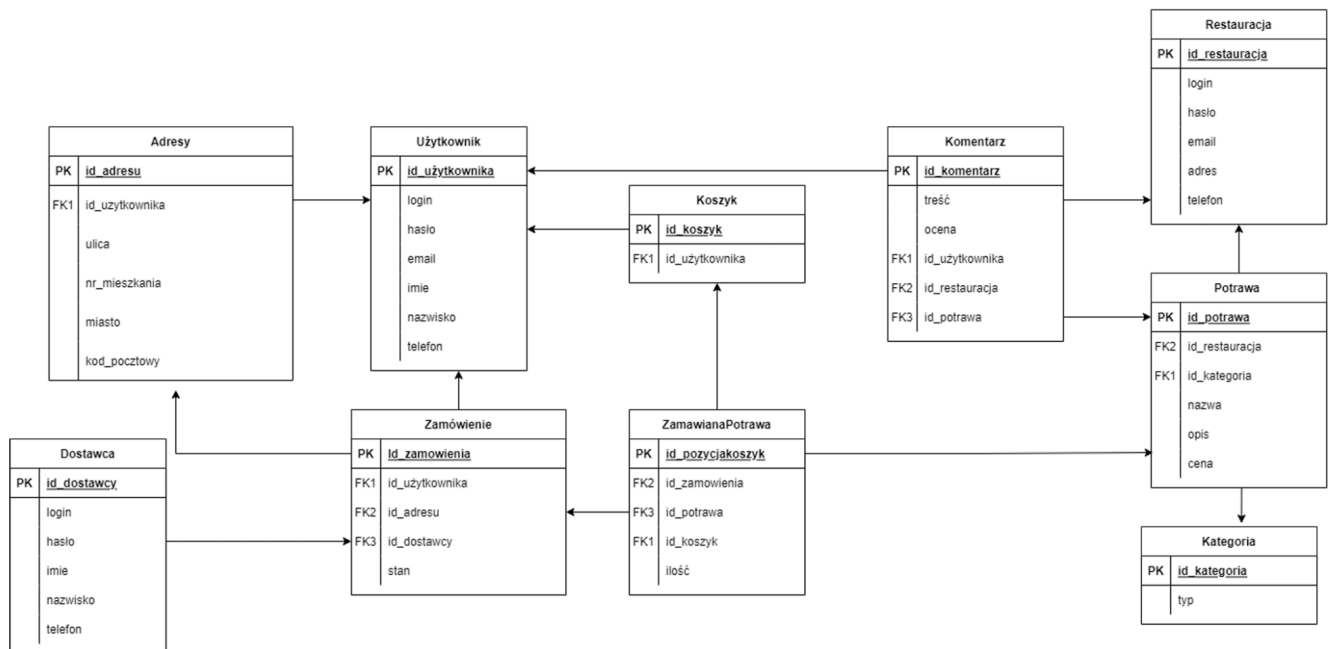
## Obsługa zamówień



# Zarządzanie restauracją



# Diagram związków encji - ERD



# Specyfikacje procesów - PSPEC

## [PSPEC 1.1: Założenie konta](#)

Wejście: Dane użytkownika

Wyjście: Nowy klient

Algorytm:

GET ID

GET hasło

If ID = PUSTE

THEN WYPISZ 'błąd, pole nie może być puste'

ELSE IF ID = NIEPOPRAWNE

THEN WYPISZ 'to nie wygląda jak identyfikator klienta, spróbuj jeszcze raz'

IF hasło = za słabe

THEN WYPISZ 'prosze zamień na mocniejsze hasło'

IF ID AND hasło = RÓWNE

THEN włącz\_przesyłanie\_rejestracji

## [PSPEC 1.2: Zalogowanie się do konta](#)

Wejście: Id\_użytkownika

Wyjście: Baza danych Użytkownicy

Algorytm:

GET ID

IF ID == wpisane ID THEN

Zalogowano pomyślnie

ELSE

Nie udało się zalogować

ENDIF

## [PSPEC 1.3: Wysyłanie wiadomości aktywacyjnej](#)

Wejście: mail użytkownika

Wyjście: Id użytkownika

Algorytm:

GET mail

IF mail = NIEPOPRAWNY

THEN WYPISZ 'niepoprawny mail'

ELSE wygeneruj\_token();

Wyślij token na danego maila

ENDIF

#### PSPEC 2.1: Dodaj użytkownika

Wejście: Dane użytkownika

Wyjście: Baza danych Użytkownicy

Algorytm:

WHILE dane użytkownika nie istnieją w bazie DO

    dodaj\_dane\_użytkownika\_do\_bazy();

END WHILE

#### PSPEC 2.2: Usuń użytkownika

Wejście: id\_użytkownika

Wyjście: Baza danych Użytkownicy

Algorytm:

WHILE użytkownik istnieje w bazie DO

    usuń\_dane\_użytkownika\_z\_bazy();

END WHILE

#### PSPEC 2.3: Dodaj restaurację

Wejście: Dane restauracji

Wyjście: Baza danych Restauracje

Algorytm:

WHILE dane restauracji nie istnieją w bazie DO

    dodaj\_dane\_restauracji\_do\_bazy();

END WHILE

#### PSPEC 2.4: Usuń restaurację

WHILE restauracja istnieje w bazie DO

    usuń\_dane\_restauracji\_z\_bazy();

END WHILE



### PSPEC 3.1: Wyświetl listę zamówień

Wejście: id restauracji

Wyjście: Baza danych Zamówienia

Algorytm:

WHILE kolejne\_zamówienia DO

    Wyświetl zamówienie

END WHILE

### PSPEC 3.2: Realizuj zamówienie

Wejście: Id Zamówienia

Wyjście: Baza danych Zamówienia

Algorytm:

WHILE kolejne\_id\_zamowienia DO

    sprawdz\_dostępność\_produktyw;

    przyrządź potrawę;

    przygotuj\_opakowanie;

    zapisz\_stan\_zamowienia;

END WHILE

### PSPEC 3.3: Aktualizuj stan zamówienia

Wejście: Id Zamówienia

Wyjście: Baza danych Zamówienia

Algorytm:

WHILE id zamówienia DO

    poinformuj\_klienta\_o\_zamówieniu;

    zaktualizuj\_stan\_zamówienia;

    zaktualizuj\_baze\_danych\_zamówienia;

END WHILE

### PSPEC 4.1: Edycja informacji o restauracji

Wejście: Informacje o restauracji

Wyjście: Baza danych Restauracje

Algorytm:

GET nowa\_informacja;

USTAW stara\_informacja NA nowa\_informacja;

### PSPEC 4.2: Dodanie potrawy

Wejście: Informacje o potrawie

Wyjście: Baza danych Potrawy

Algorytm:

WHILE dane\_potrawy\_nie\_istnieją\_w\_bazie DO

    dodaj\_dane\_potrawy\_do\_bazy();

END WHILE

#### PSPEC 4.3: Usuwanie potrawy

Wejście: Id potrawy

Wyjście: Baza danych Potrawy

Algorytm:

WHILE dane potrawy istnieją w bazie DO

usuń\_dane\_potrawy\_z\_bazy();

END WHILE

#### PSPEC 4.4: Edycja potrawy

Wejście: Informacje o potrawie

Wyjście: Baza danych Potrawy

Algorytm:

GET nowa\_potrawa;

GET opis\_nowej\_potrawy;

stara\_potrawa = nowa\_potrawa

stary\_opis\_potrawy = nowy\_opis\_potrawy;

#### PSPEC 5.1: Dodaj do koszyka

Wejście: Potrawa

Wyjście: Baza danych Koszyk

Algorytm:

WHILE liczba produktów nie przekracza zapasu danej potrawy w danej restauracji  
DO

dodaj\_produkt\_do\_koszyka;

END WHILE

zaktualizuj bazę danych Koszyk dla danego id użytkownika;

#### PSPEC 5.2: Usuń z koszyka

Wejście: id użytkownika, id produktu / potrawy

Wyjście: baza danych Koszyk

Algorytm:

GET nazwa\_produktu;

Usuń nazwa\_produktu z koszyka id użytkownika;

Zaktualizuj bazę danych Koszyk;

#### PSPEC 5.3: Złóż zamówienie

Wejście: Id klienta, id potrawy

Wyjście: Baza danych Zamówienia

Algorytm:

WHILE id potrawy > 0 DO

wyślij prośbę o realizację zamówienia do bazy danych Zamówienia z daną  
potrawą;

id potrawy--;

END WHILE

#### PSPEC 6.1:Komentarz do restauracji

Wejście: Komentarz, nazwa\_restauracji, nazwa\_potrawy

Wyjście: Baza danych Komentarze

Algorytm:

GET komentarz, nazwa\_restauracji;

SET komentarz DLA nazwa\_restauracji;

#### PSPEC 6.2:Komentarz do potrawy

Wejście: Komentarz, nazwa\_restauracji, nazwa\_potrawy

Wyjście: Baza danych Komentarze

Algorytm:

GET komentarz, nazwa\_restauracji, nazwa\_potrawy;

SET komentarz DLA id\_potrawy DLA id\_restauracji;