

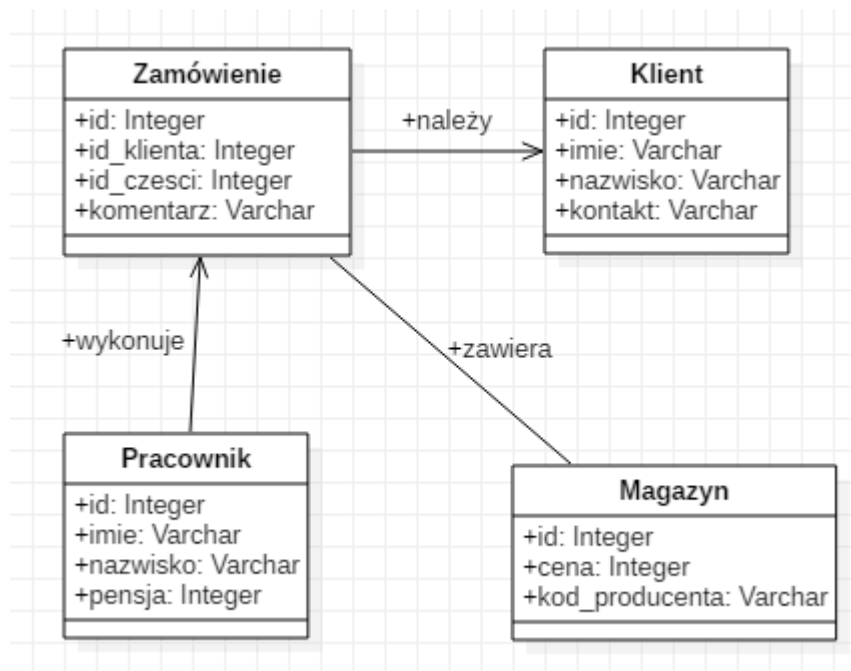
# Serwis komputerowy

Jakub Bojarski

Projekt UML aplikacji serwisu komputerowego, w którym dany serwis trzyma informacje o klientach i ich zamówieniach, historię zamówień (tych zrealizowanych i niezrealizowanych).

Aplikacja składa się z samej aplikacji okienkowej napisanej w Javie oraz bazy danych (OracleDB).

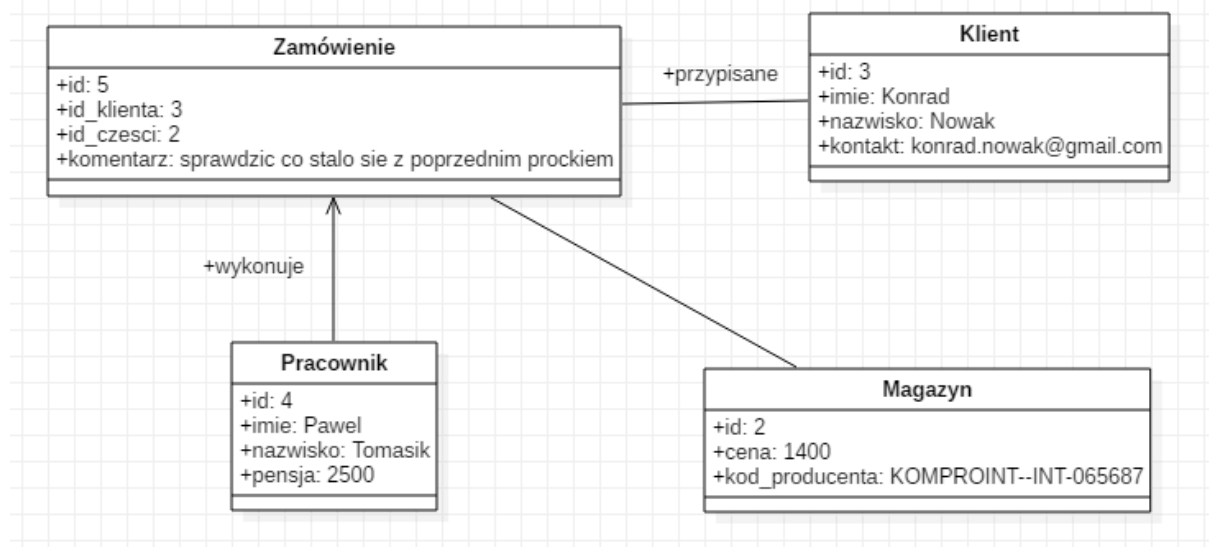
Diagram Klas:



- `id_klienta` to id z klasy **Klient**;
- `id_czesci` to id z klasy **Magazyn**;
- Pole `kontakt` u klienta powinno zawierać e-mail/numer telefonu bądź inny sposób na komunikację z klientem.
- Pole `kod_producenta` w magazynie zawiera kod, który producent przypisał danej części.

Osoba obsługująca aplikację może dodać pracownika oraz nowe zamówienie dla istniejącego lub nowego klienta podając przy tym wszystkie niezbędne dane.

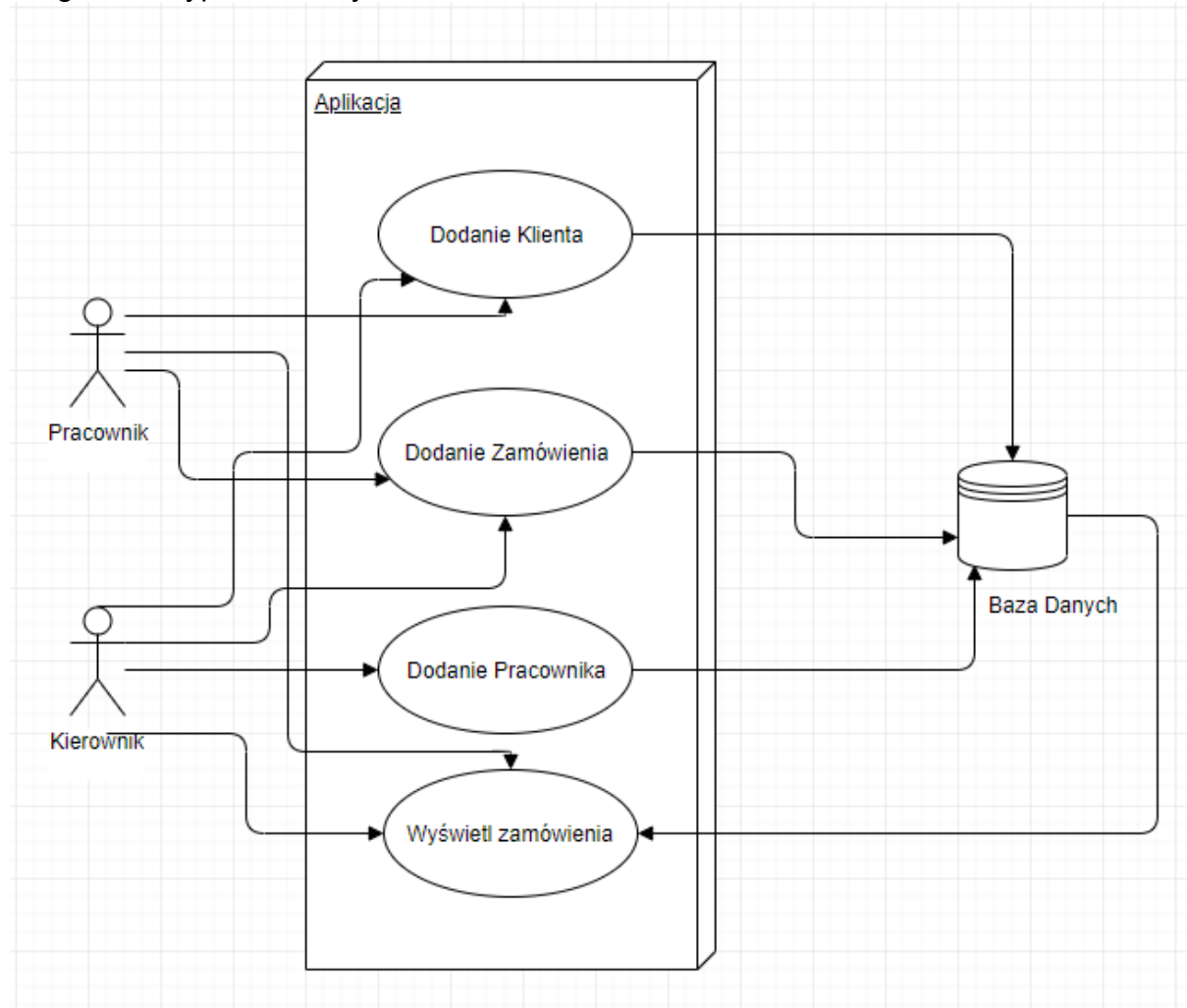
## Diagram Obiektów:



Obiekty w bazie są zapisywane tak jak klasy, ale w momencie gdy potrzeba, np. wyświetlić dane o zamówieniu to dzięki id\_klienta można bez problemu połączyć zamówienie z klientem i wyświetlić wszystkie dane, bez redundancji informacji trzymanych w samej bazie.

Na powyższym przykładzie klient Konrad Nowak złożył zamówienie na nowy procesor o kodzie KOMPROINT--INT-065687 za cenę 1400 PLN. Zamówienie będzie obsługiwał Paweł Tomasik. Jednak klient nie jest pewny co się stało poprzedniemu procesorowi, więc poprosił by osoba tworząca zlecenie dodała w komentarzu, żeby pracownik sprawdził co się stało.

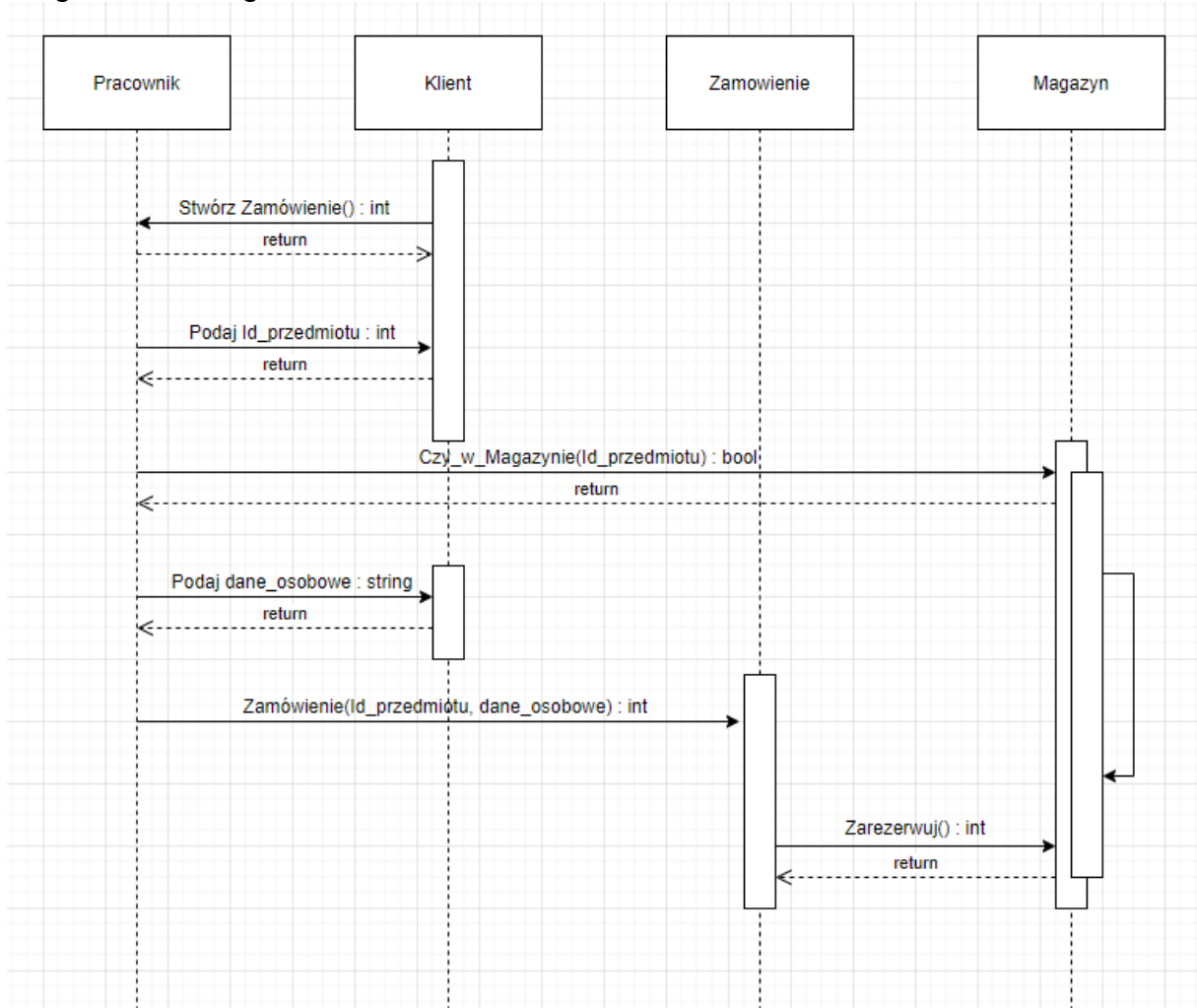
## Diagram Przypadków użycia:



Tylko kierownik może dodawać pracowników, ale wszyscy mogą dodawać klienta, zamówienie czy wyświetlać zamówienia.

- Dodanie Klienta: Otwiera okienko z polami których wpisujemy potrzebne dane, aby dodać klienta;
- Dodanie Zamówienia: Otwiera okienko z polami w których wpisujemy potrzebne dane, aby dodać nowe zamówienie;
- Dodanie Pracownika: Otwiera okienko z polami w których wpisujemy potrzebne dane, aby dodać pracownika;
- Wyświetl zamówienia wysyła prośbę do bazy danych i w odpowiedzi wyświetlane są wszystkie zamówienia;

## Diagram Przebiegu:



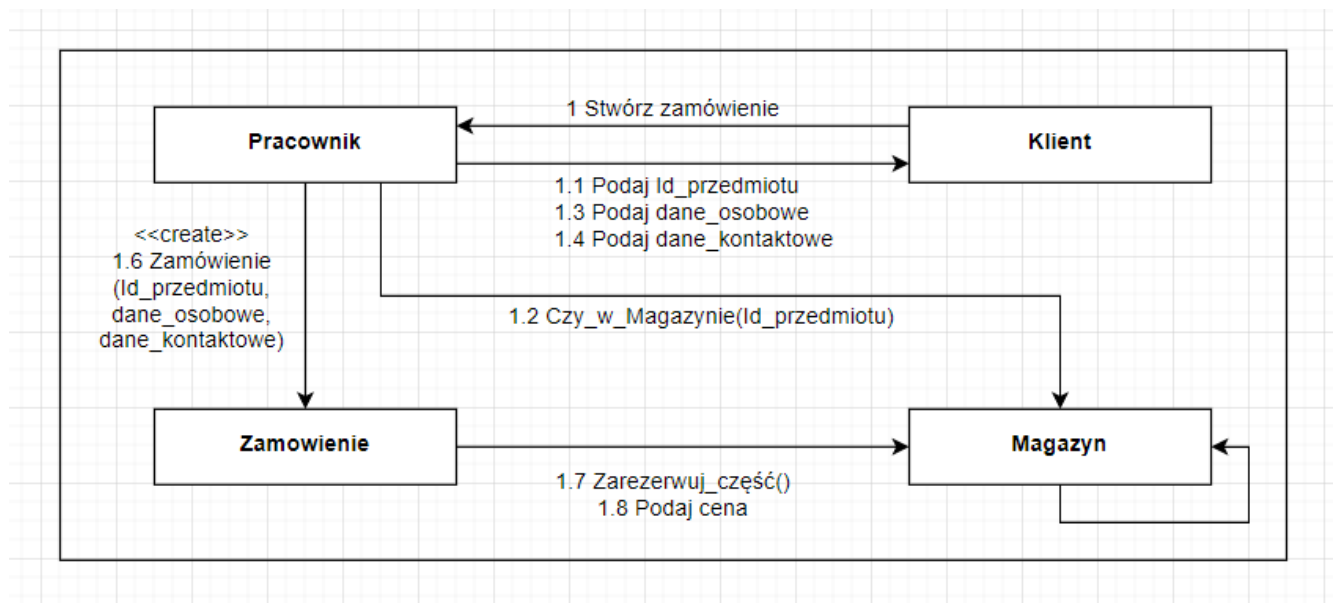
Każda funkcja stwórz lub podaj, wysła zapytanie o informacje zwrótną w odpowiedniej formie.

Funkcja Czy\_w\_magazynie ma jedynie zwrócić prawdę jeśli towar jest lub fałsz jeśli nie ma.

Zamówienie gromadzi dane i zapisuje je.

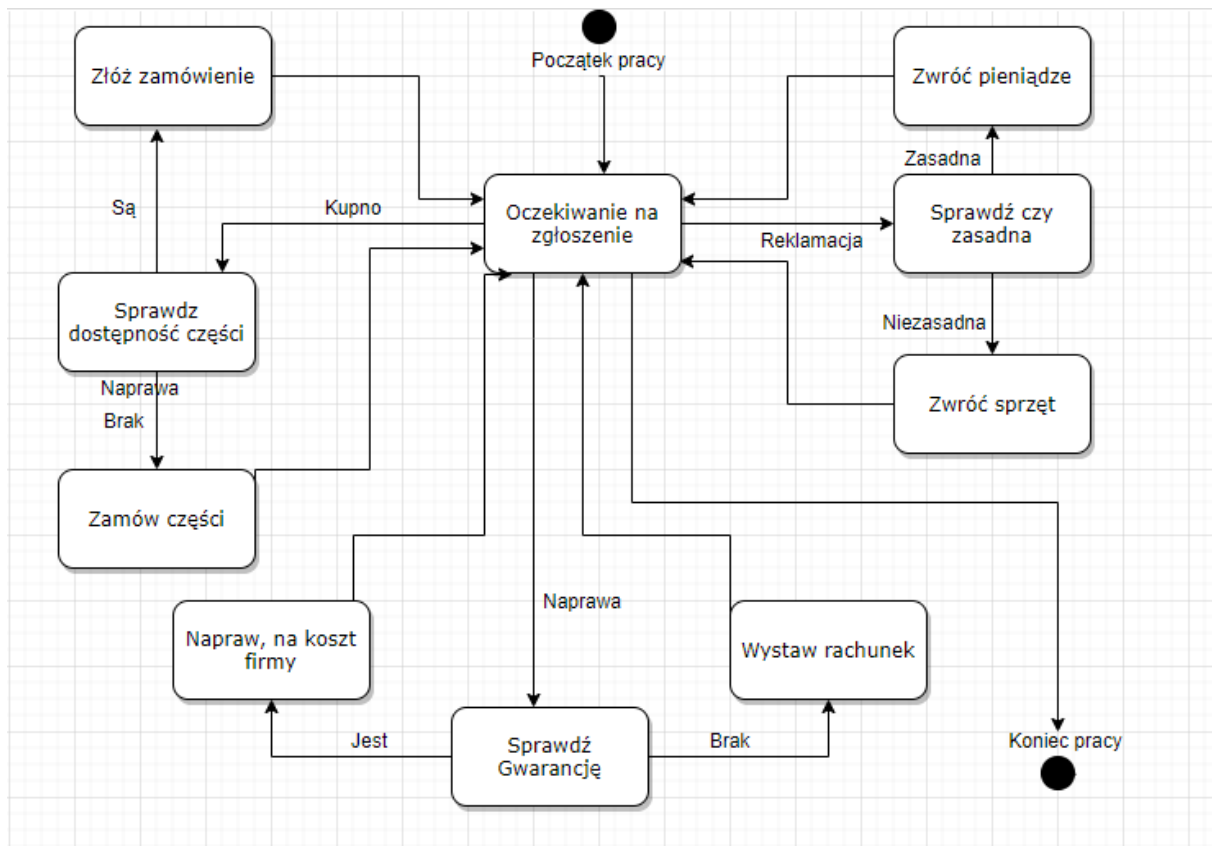
Zarezerwuj wysła sygnał do magazynu aby przedmiot z tego zamówienia nie był już dostępny.

## Diagram Kooperacji:



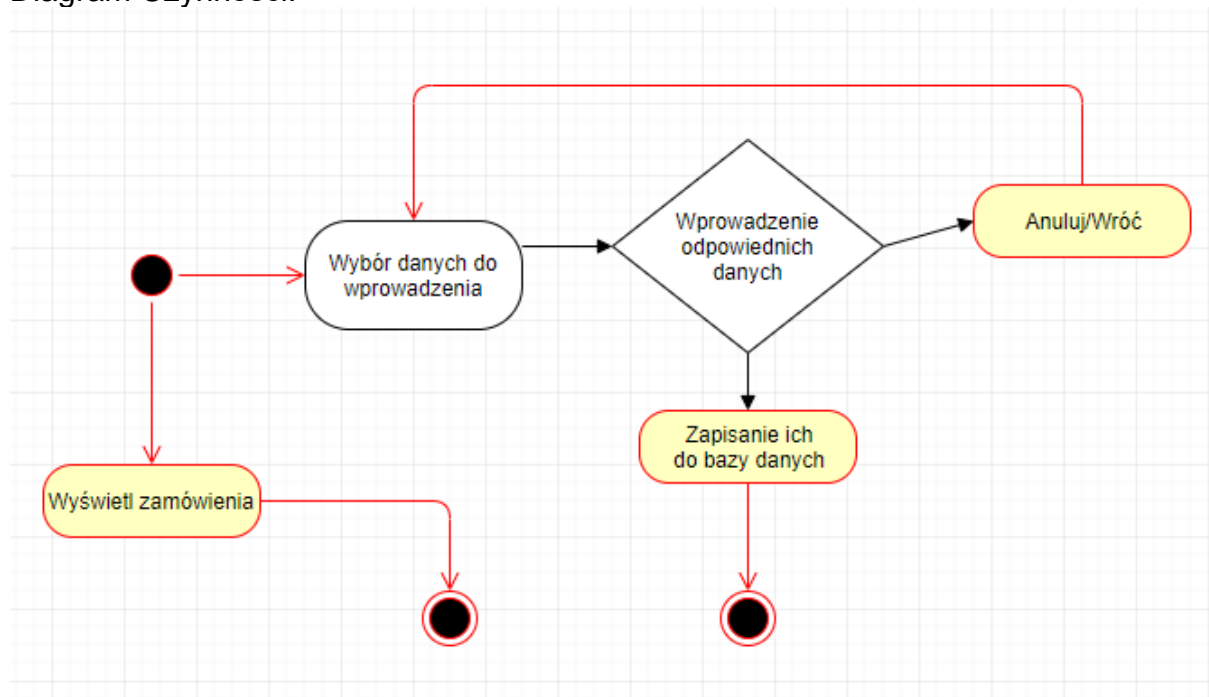
Kooperacja zachodzi pomiędzy wszystkimi w Bazie, kiedy klient prosi pracownika o złożenie zamówienia ten sprawdza najpierw czy w magazynie znajduje się przedmiot, o który klient prosi. Następnie wprowadza dane do Zamówienia i rezerwuje i finalizuje je ceną.

## Diagram Stanu:



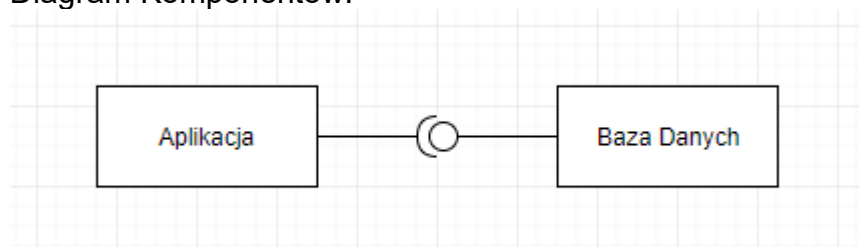
Stan aplikacji zależy od typu zgłoszenia, przykładowo Kupno, Reklamacja, Gwarancja i zwraca różne wartości w zależności od sytuacji. Kiedy zostanie zrealizowane aktualne zgłoszenie aplikacja czeka na kolejne, aż do zakończenia pracy.

### Diagram Czynności:



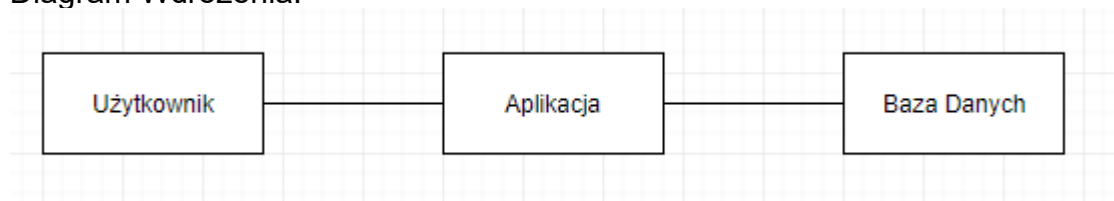
Wybór danych do wprowadzenia to po prostu wybór czy ktoś chce wprowadzić nowego klienta, zamówienie czy pracownika. Dodawanie nowego obiektu można anulować i powrócić do ekranu wyboru danych do wprowadzenia. Można również wyświetlić już istniejące zamówienia.

### Diagram Komponentów:



Pamiętajmy, że jest to aplikacja, która tylko pomaga firmie zachować czytelność danych o pracownikach, klientach i zamówieniach; aplikacja ta nie ma zbyt wielu funkcji i nie potrzebuje do działania zbyt wielu komponentów.

### Diagram Wdrożenia:



Aplikacja ma dostęp do serwera firmy na którym stoi baza danych.

Diagramy zrobione przy pomocy: <https://www.draw.io/> oraz staruml(2 pierwsze).

