



**Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot :**  
**Inżynieria Oprogramowania**

**Temat projektu :**  
***System zarządzania zamówieniami***

**Wykonali :**  
**Karol Bury, Piotr Długosz, Kamil Filar, Patryk Dyndał**

**Prowadzący: dr inż. Maksymilian Knap**

### Spis treści:

1. Opis realizowanego systemu .....	3
2. Opis zakładanej funkcjonalności .....	3
3. Opis api .....	3
4. Diagram ERD i opis .....	4
5. Algorytm wystawienia nowego zamówienia .....	7
6. Opis wykorzystanych technologii .....	9
7. Funkcjonalności .....	9
8. Opis zrealizowanej aplikacji – prezentacji aplikacji .....	10
9. Testy akceptacyjne .....	12
10. Opis instalacji aplikacji .....	13

# 1 Opis realizowanego systemu

System ma na celu automatyczne koordynowanie zamówień według odpowiednio przygotowanych procesów biznesowych. System jest przystosowany pod obsługę przez pracownika, który będzie mógł manualnie obsługiwać dane zamówienie (edytować je), modyfikować status zamówienia w razie zaistnienia takiej potrzeby oraz wprowadzać/usuwać dane zamówienia. W obsłudze tego systemu pomoże mu intuicyjna wyszukiwarka z filtrowaniem wyników.

# 2 Opis zakładanej funkcjonalności

- Podgląd danego zamówienia - System pobiera dane zamówienia z tabeli Order\_Details w którym znajduje się lista produktów, dane kontrahenta i dodatkowe informacje o zamówieniu.
- Manualna modyfikacja zamówienia - System pozwala modyfikować listę produktów danego zamówienia. Przyjmujemy że, jeżeli klient będzie chciał stworzyć zamówienie na inne dane, to lista produktów musi zostać utworzona na nowo.
- Modyfikacja statusu zamówienia - Zmiana statusu zamówienia na ("Zamówiono", "W trakcie", "Zrealizowano").
- Wyszukiwanie zamówień wraz z filtrowaniem wyników
- Wprowadzanie zamówienia - Ręczne wprowadzanie w systemie zamówień podając dane kontrahentów oraz listę produktów.
- Usuwanie zamówień

# 3 Opis api

- Lista wszystkich requestów wraz z przykładowymi danymi wysyłanymi, zwracanymi znajduje się pod linkiem: <https://documenter.getpostman.com/view/12136197/TVzVhbHh>

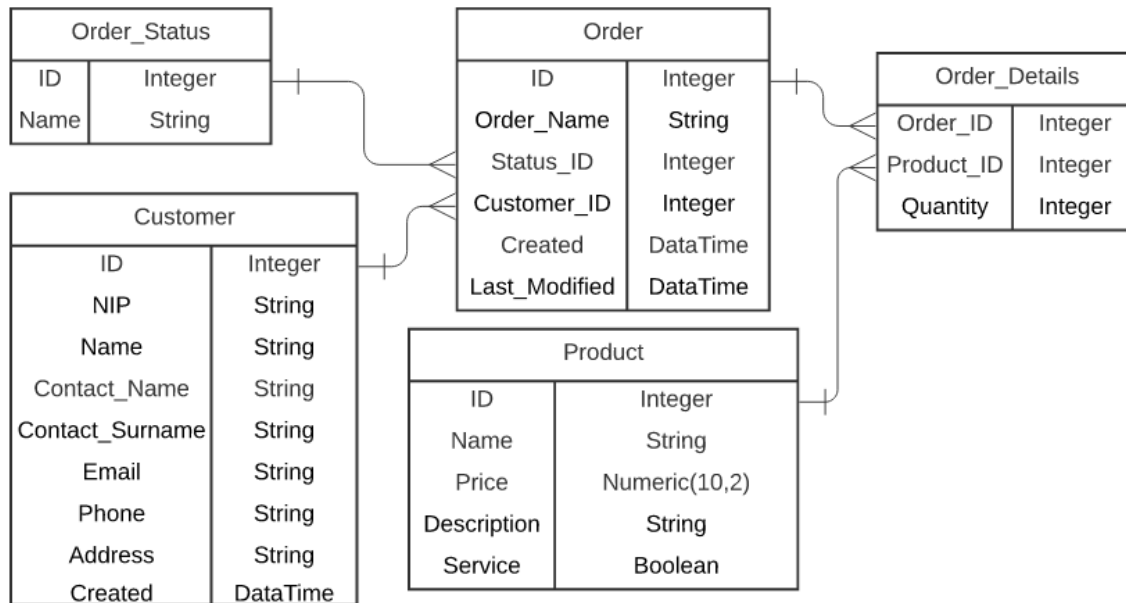
## 4 Struktura Obiektu zamówienia

Każde zamówienie jest zapisywane w postaci struktury tego typu która jest pobierana z api po id zamówienia.

```
{
  "id": 2,
  "order_name": "ZM/2021/1/15/NR/1",
  "status_id": 1,
  "customer_id": 2,
  "created_at": "2021-01-15T18:05:45.000000Z",
  "updated_at": "2021-01-15T18:05:45.000000Z",
  "customer": {
    "id": 2,
    "NIP": "1234",
    "name": "nazwa firmy"
  },
  "status": {
    "id": 1,
    "name": "Zmówiono"
  },
  "products": [
    {
      "id": 1,
      "product_id": 1,
      "name": "nazwa produktu",
      "vatRate": 1.23,
      "measureUnit": "kg",
      "isService": 0,
      "pivot": {
        "order_id": 2,
        "product_id": 1,
        "quantity": 2,
        "netPrice": "30.54"
      }
    },
    {
      "id": 2,
      "product_id": 2,
      "name": "nazwa produktu 2",
      "vatRate": 1.05,
      "measureUnit": "szt",
      "isService": 1,
      "pivot": {
        "order_id": 2,
        "product_id": 2,
        "quantity": 20,
        "netPrice": "186.39"
      }
    }
  ]
}
```

- order\_name – numer zamówienia (generowany z daty i ilości zamówień danego dnia)
- customer – informacje odnośnie kontrahenta / klienta (tylko podstawowe)
- status – aktualny status zamówienia
- products – lista produktów w zamówieniu
- pivot – tabela łącząca produkt z zamówieniem (w niej przechowujemy cenę, ilość)
- vatRate – stawka Vat stała dla każdego produktu

## 5 Diagram ERD i opis encji



- Order – zamówienia**

Kolumna	Typ	Opis
ID	Integer	Główne Id tabeli
Order_Name	String	Numer Zamówienia ("ZM/2020/11/4/NR/1")
Status_ID	Integer	Identyfikator statusu
Customer_ID	Integer	Identyfikator klienta
Created	DateTime	Data utworzenia zamówienia
Last_Modified	DateTime	Data ostatniej modyfikacji zamówienia

- Customer – klienci / kontrahenci**

Kolumna	Typ	Opis
ID	Integer	Główne Id tabeli
NIP	String	Numer identyfikacji podatkowej
Name	String	Nazwa firmy
Contact_Name	String	Imię osoby kontaktowej
Contact_Surname	String	Nazwisko osoby kontaktowej
Email	String	E-mail
Phone	String	Telefon
Address	String	Adres klienta
Created	DateTime	Data dołączenia

- **Product – produkty**

Kolumna	Typ	Opis
ID	Integer	Id zamówienia
Name	String	Nazwa produktu
Price	Numeric(10,2)	Cena netto
Description	String	Opis produktu
Service	Boolean	Czy produkt jest usługą

- **Order\_Details – szczegóły zamówieniami**

Kolumna	Typ	Opis
Order_ID	Integer	Id zamówienia
Product_ID	Integer	Id produktu
Quantity	Integer	Ilość danego produktu

- **Order\_Status**

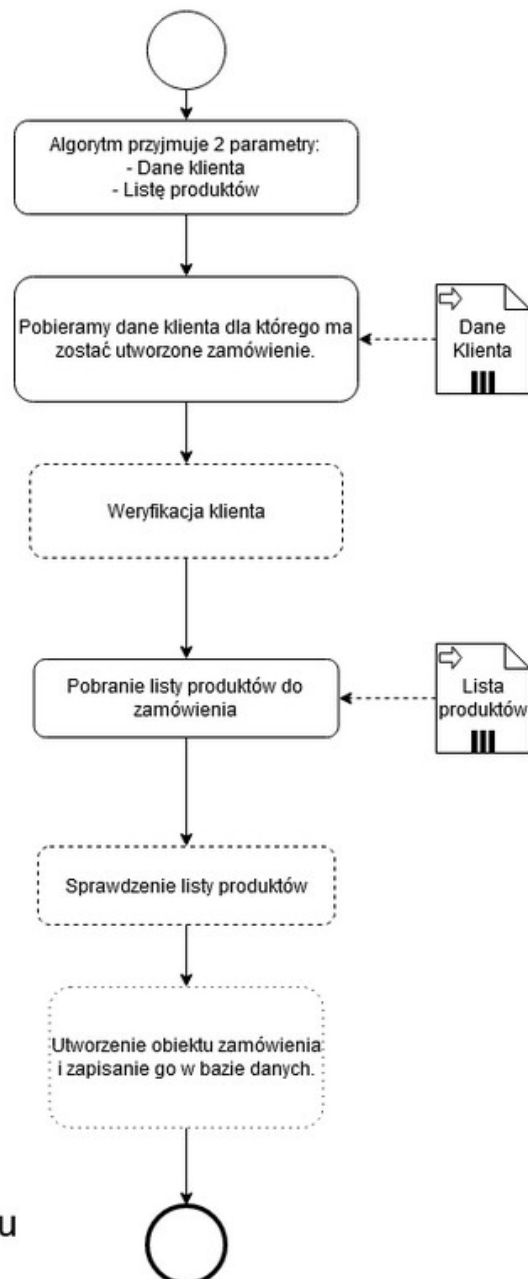
Kolumna	Typ	Opis
ID	Integer	Główne Id tabeli
Name	String	Nazwa aktualnego stanu zamówienia ("Zamówiono", "W trakcie", "Zrealizowano")

## 6 Algorytm wystawienia nowego zamówienia

Metoda przyjmuje dwa parametry wejściowe:  
Lista produktów z dodatkowym polem odnośnie ilości  
Dane Klienta

Sprawdzamy czy istnieje dany produkt u nas w bazie danych ?

Zwrócenie obiektu  
Zamówienia



### Przypisanie produktów do zamówienia

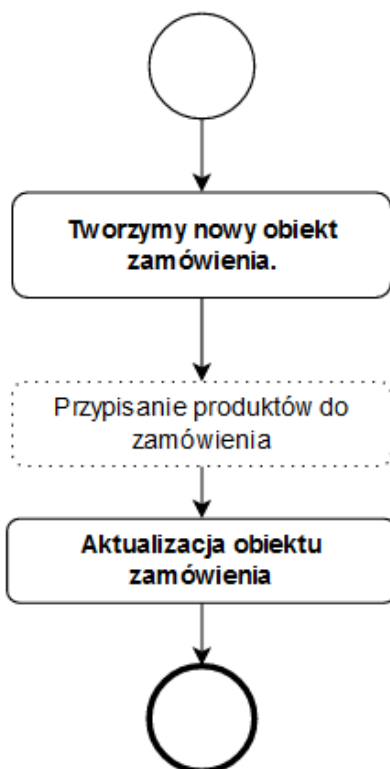


Każdy produkt znajdujący się na liście posiada dodatkowy parametr odnośnie ilości ( tym się różni od zwykłego obiektu produktu )

Dla każdego produktu z listy tworzymy nowy obiekt **"Order\_Details"**  
**"Order\_ID"** - przypisujemy id obiektu zamówienia utworzonego chwilę wcześniej  
**"Product\_ID"** - przypisujemy id produktu  
**"Quantity"** - przypisujemy dodatkowy parametr obiektu produkt ( ilość )

Aktualizacja obiektu zamówienia **"Order"**:  
**"Last\_Modified"** - aktualizacja daty ostatniej modyfikacji

### Utworzenie obiektu zamówienia i zapisanie go w bazie danych.

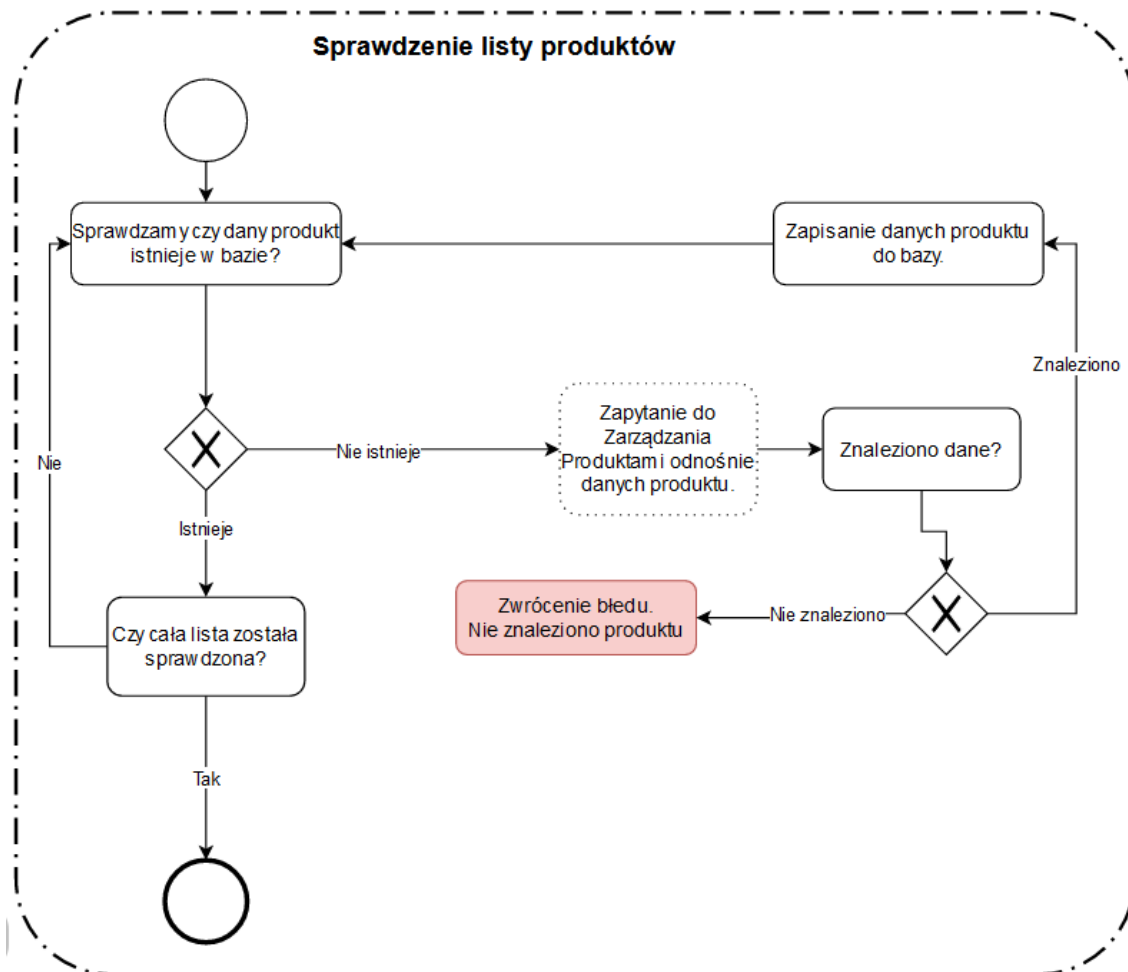


Tworzenie nowego obiektu zamówienia **"Order"** :  
**"ID"** - automatycznie przez bazę  
**"Order\_Name"** - generowany z aktualnej daty, ilości zamówień danego dnia  
**"Status\_ID"** - pusty  
**"Customer\_ID"** - przypisanie id klienta  
**"Created"** - automatycznie dodana data utworzenia  
**"Last\_Modified"** - automatycznie dodana data modyfikacji

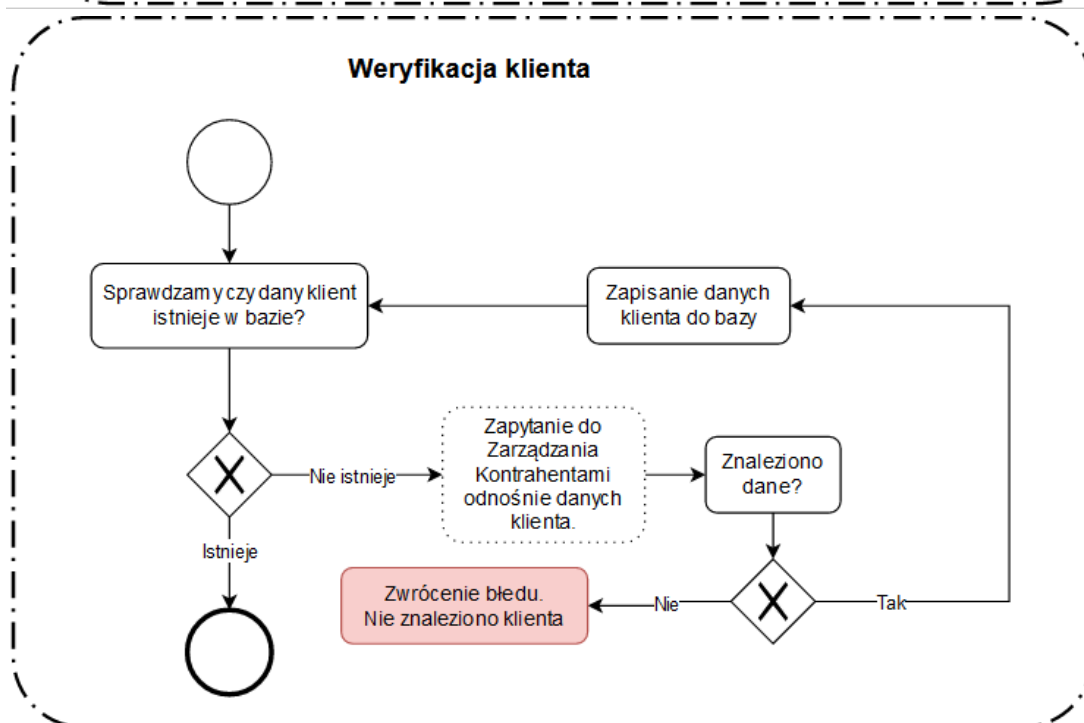
Aktualizacja obiektu zamówienia **"Order"**:  
**"Status\_ID"** - przypisanie statusu **"Zamówiono"**  
**"Last\_Modified"** - aktualizacja daty ostatniej modyfikacji



### Sprawdzenie listy produktów



### Weryfikacja klienta



## **7 Opis wykorzystanych technologii**


- Framework Laravel dla PHP – wykorzystane do obsługi backendy, połączenia z bazą.
- Framework Angular – wykorzystany do frontendu.
- Baza danych MySQL – wykorzystana do przechowywania danych.

## **8 Funkcjonalności**




- Podgląd danego zamówienia - System pobiera dane zamówienia z tabeli Order\_Details w którym znajduje się lista produktów, dane kontrahenta i dodatkowe informacje o zamówieniu.
- Manualna modyfikacja zamówienia - System pozwala modyfikować listę produktów danego zamówienia. Przyjmujemy że, jeżeli klient będzie chciał stworzyć zamówienie na inne dane, to lista produktów musi zostać utworzona na nowo.
- Modyfikacja statusu zamówienia - Zmiana statusu zamówienia na ("Zamówiono", "W trakcie", "Zrealizowano").
- Wyszukiwanie zamówień wraz z filtrowaniem wyników
- Wprowadzanie zamówienia - Ręczne wprowadzanie w systemie zamówień podając dane kontrahentów oraz listę produktów.
- Usuwanie zamówień

## 9 Opis zrealizowanej aplikacji – prezentacji aplikacji


- Widok z listy zamówień

 System zarządzania zamówieniami






[Lista zamówień](#)[Lista produktów](#)[Lista kontrahentów](#)[Zamówienie](#)

Lp.	Nazwa zamówienia	NIP Kontrahenta	Status
1	ZM/2021/1/19/NR/1	1234	Ukończono  
2	ZM/2021/1/19/NR/2	12345678901	Zamówiono  
3	ZM/2021/1/19/NR/3	12345678901	W realizacji  

- Widok z listy produktów

 System zarządzania zamówieniami

[Lista zamówień](#)[Lista produktów](#)[Lista kontrahentów](#)[Zamówienie](#)

Lp.	Nazwa produktu	Cena netto (PLN)	VAT	Jednostka
1	asperiores	30.54	1.23	Harum autem nihil fugit ut vel omnis atque.  <a href="#">Szczegóły</a>
2	ea	186.39	1.05	Mollitia praesentium voluptatem est distinctio vel.  <a href="#">Szczegóły</a>
3	iste	297.11	1.05	Vitae nisi consequatur corrupti error nesciunt exercitationem.  <a href="#">Szczegóły</a>
4	laudantium	467.36	1.23	In voluptatem nihil nesciunt illum.  <a href="#">Szczegóły</a>
5	provident	193.32	1.08	Doloremque mollitia sint quis sint.  <a href="#">Szczegóły</a>

- Widok tworzenia zamówieniami

 System zarządzania zamówieniami

[Lista zamówień](#)[Lista produktów](#)[Lista kontrahentów](#)[Zamówienie](#)

 Tworzenie zamówienia:

NIP: 12345678901  
Nazwa firmy: kontrahent2

[Wybierz kontrahenta](#)[Szczegóły kontrahenta](#)

Lp	Nazwa produktu	Cena netto	Cena brutto	Ilość	Usun
1	asperiores	30.54	37.56	 1 	
2	laudantium	467.36	574.85	 1 	

Całkowita cena netto: 497.90  
Całkowita cena brutto: 612.42

[← Anuluj](#)[✓ Zamów](#)

- Widok informacji o zamówieniu

Informacje o zamówieniu

NIP: 12345678901

Nazwa firmy: kontrahent2

Stworzony: Tue Jan 19 2021

Edytowany: Tue Jan 19 2021

Szczegóły kontrahenta

Lp.	Nazwa produktu	Cena netto	Cena brutto	Ilość	Usuń
1	nazwa produktu 2	30.54	32.07	<div>+</div> 1 <div>-</div>	<div></div>
2	laudantium	467.36	574.85	<div>+</div> 1 <div>-</div>	<div></div>

+

 Dodaj przedmiot

Status: 

Zamówiono

Całkowita cena netto: 497.90 zł

Całkowita cena brutto: 606.92 zł

←

 Powrót

✓

 Zapisz

- Widok listy kontrahentów

System zarządzania zamówieniami

Lista zamówień

Lista produktów

Lista kontrahentów

Zamówienie

Lp.	NIP	Nazwa	
1	1234	nazwa firmy	<div></div>
2	12345678901	kontrahent2	<div></div>

- Widok informacji o kontrahencie

System zarządzania zamówieniami

Lista zamówień

Lista produktów

Lista kontrahentów

Zamówienie

Szczegóły kontrahenta

NIP: 12345678901

Nazwa firmy: kontrahent2

Adres: adres

Dane kontaktowe:

Imię: imie

Nazwisko: nazwisko

Email: email@email.com

Telefon: tel

Lp.	Nazwa zamówienia	Status	
1	ZM/2021/1/19/NR/2	Zamówiono	<div></div>
2	ZM/2021/1/19/NR/3	W realizacji	<div></div>

←

 Powrót

+

 Dodaj zamówienie

# 10 Testy akceptacyjne

## Frontend :

```
> front@0.0.0 build D:\3 rok\IO\IO_System_Zarzadzania_Zamowieniami\front
> ng build

✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Names | Size
vendor.js           | vendor | 5.11 MB
main.js             | main   | 202.56 kB
polyfills.js        | polyfills | 141.31 kB
styles.css          | styles | 77.52 kB
runtime.js          | runtime | 6.15 kB
                    | Initial Total | 5.53 MB

Build at: 2021-01-19T18:31:38.563Z - Hash: ec19107acfd1181d5f9a - Time: 19361ms

Warning: D:\3 rok\IO\IO_System_Zarzadzania_Zamowieniami\front\src\app\components\product-list\product-list.component.ts depends on 'lodash'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

Terminal will be reused by tasks, press any key to close it.
```

```
▼ {created_at: "2021-01-19T18:36:18.000000Z", order_name: "ZM/2021/1/19/NR/2", status_id: 1, customer_id: 1, updated_at: "2021-01-19T18:36:18.000000Z", ...} cart.component.ts:99
  created_at: "2021-01-19T18:36:18.000000Z"
  customer_id: 1
  id: 3
  order_name: "ZM/2021/1/19/NR/2"
  status_id: 1
  updated_at: "2021-01-19T18:36:18.000000Z"
  __proto__: Object
```

## Backend:

```
D:\3 rok\IO\IO_System_Zarzadzania_Zamowieniami\OrderManagementSystem>php artisan migrate:fresh --seed
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table (56.43ms)
Migrated: 2014_10_12_000000_create_users_table (56.43ms)
Migrating: 2014_10_12_100000_create_password_resets_table (177.68ms)
Migrated: 2014_10_12_100000_create_password_resets_table (177.68ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table (70.29ms)
Migrated: 2019_08_19_000000_create_failed_jobs_table (70.29ms)
Migrating: 2020_11_25_154024_order_status (25.16ms)
Migrated: 2020_11_25_154024_order_status (25.16ms)
Migrating: 2020_11_25_154733_customers (27.19ms)
Migrated: 2020_11_25_154733_customers (27.19ms)
Migrating: 2020_11_25_154846_orders (199.80ms)
Migrated: 2020_11_25_154846_orders (199.80ms)
Migrating: 2020_11_25_154856_products (32.09ms)
Migrated: 2020_11_25_154856_products (32.09ms)
Migrating: 2020_11_27_185446_create_order_product_pivot_table (504.43ms)
Migrated: 2020_11_27_185446_create_order_product_pivot_table (504.43ms)
Database seeding completed successfully.

D:\3 rok\IO\IO_System_Zarzadzania_Zamowieniami\OrderManagementSystem>php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Tue Jan 19 19:43:16 2021] PHP 7.4.10 Development Server (http://127.0.0.1:8000) started
[Tue Jan 19 19:43:24 2021] 127.0.0.1:54498 Accepted
[Tue Jan 19 19:43:24 2021] 127.0.0.1:54498 Closing
[Tue Jan 19 19:43:26 2021] 127.0.0.1:54498 Accepted
[Tue Jan 19 19:43:27 2021] 127.0.0.1:54498 Closing
[Tue Jan 19 19:43:30 2021] 127.0.0.1:54507 Accepted
[Tue Jan 19 19:43:30 2021] 127.0.0.1:54507 Closing
[Tue Jan 19 19:43:31 2021] 127.0.0.1:54514 Accepted
[Tue Jan 19 19:43:32 2021] 127.0.0.1:54514 Closing
```

```
D:\3 rok\IO\IO_System_Zarzadzania_Zamowieniami\OrderManagementSystem>php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\OrderTest
✓ create order
✓ create order without seeded db
✓ create order with date
✓ create order missing data
✓ get orders
✓ get order by id
✓ get unexisted order by id
✓ update order
✓ update unexisted order
✓ update order invalid data
✓ delete order
✓ delete unexisted order

Tests: 12 passed
Time: 3.12s
```

## 11 Opis instalacji aplikacji

Projekt znajduje się na publicznym repozytorium git:

[https://github.com/dydzia04/IO\\_System\\_Zarzadzania\\_Zamowieniami](https://github.com/dydzia04/IO_System_Zarzadzania_Zamowieniami) należy go pobrać.

### **Zalecana opcja instalacji aplikacji docker :**

1. docker-compose build
2. docker-compose up -d

**Następnie na obrazie dockera z laravelem należy wykonać:**

3. npm run clearDBandSeed
4. npm run runApp

### **Druga opcja instalacji:**

**Uruchomienie frontendu w katalogu „frontend”**

1. npm install
2. npm install @angular/cli
3. ng build
4. ng serve

**Uruchomienie backendu w katalogu „OrderManagementSystem”**

Domyślnie backend jest skonfigurowany aby działać z bazą danych MySql na porcie 3306.

Dane odnośnie bazy danych zmieniamy w pliku „.env” plik „.env” otrzymujemy kopiując go z pliku „.env.example”

**Dane do zmiany :**    „ DB\_PORT=3306

DB\_DATABASE=ordermanagementsystem

DB\_USERNAME=ordermanagementsystem

DB\_PASSWORD=ordermanagementsystem”

Dane możemy pozostawić takie same lecz należy na serwerze MySql utworzyć konto „ordermanagementsystem” z hasłem „ordermanagementsystem” i bazą danych „ordermanagementsystem”.

Następnie należy wykonać polecenia.

1. composer install
2. npm install
3. npm run clearDBandSeed
4. npm run runApp