

# Bazy danych 1

## Dokumentacja do projektu

Temat projektu:

Obsługa sklepu internetowego

Viacheslav Bylo  
Informatyka stosowana  
Wydział Fizyki i Informatyki Stosowanej  
Akademia Górniczo-Hutnicza w Krakowie

## I. Projekt koncepcji, założenia

### 1. Zdefiniowanie tematu projektu

Tematem projektu jest stworzenia serwisu dla sklepu internetowego, który pozwala klientowi kupić wybrany produkt z wybranej kategorii.

### 2. Analiza wymagań użytkownika

Do aplikacji można się zalogować jako: administrator lub klient.

Po zalogowaniu klient będzie mógł zobaczyć wszystkie dostępne kategorie produktów oraz same produkty, dodać do listy zamówień to co będzie chciał kupić, a na końcu złożyć zamówienie. Także, użytkownik będzie mógł zmienić swój adres, do którego będą wysłane zamówione produkty.

Po zalogowaniu ze strony menedżera, można będzie zobaczyć całą bazę, to znaczy że będą widoczne tabele z zamówieniami, produktami, magazynami, nadawcami oraz spedytorami. Elementy w każdej z tych tabel można zmieniać. Na przykład: wysłać zamówienie, dodać lub usunąć produkt, magazyn, nadawcę lub spedytora.

Baza danych umożliwia odczytywanie wartości z tabel i wprowadzanie nowych.

### 3. Zaprojektowanie funkcji

- **Logowanie** - logowanie do systemu (jako klient lub menedżer).

**Klient:**

- **Tworzenie konta** – możliwość założenia konta dla nowych pacjentów.
- **Wybór kategorii** - wybranie kategorii produktów w której jest zainteresowany.
- **Wybór produktu oraz dodanie go na listę** - wybranie produktów które chce kupić.
- **Zmiana adresu wysłania.**

## Menedżer:

- Wysyłanie zamówień
- Dodawanie oraz usuwanie produktów
- Dodawanie oraz usuwanie magazynów
- Dodawanie oraz usuwanie nadawców
- Dodawanie oraz usuwanie spedytorów

## II. Projekt diagramów

### 4. Budowa diagramu ERD

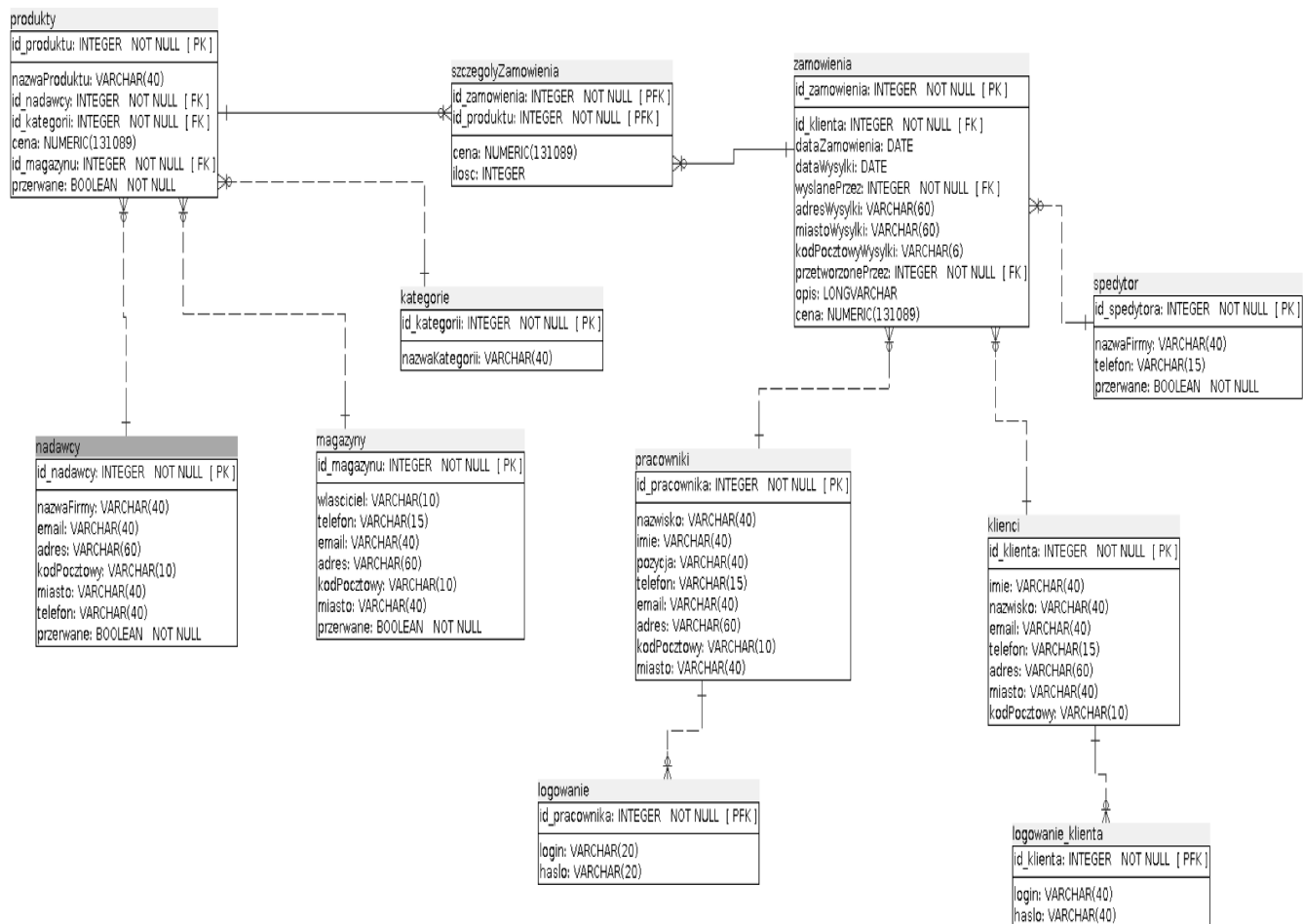
Występują zarówno relacje 1:N, jak i N:M. Tabele powiązane relacjami N:M mają między sobą tabele asocjacyjne.

### 5. Zdefiniowanie encji (obiektów) oraz ich atrybutów

W stworzonej bazie danych istnieją poniższe encje:

- **produkty** - id\_produktu, nazwaProduktu, id\_nadawcy, id\_kategorii, cena, id\_magazynu, przerwane
- **magazyny** - id\_magazynu, wlasciciel, telefon, email, adres, kodPocztowy, miasto, przerwane
- **nadawcy** - id\_nadawcy, nazwaFirmy, email, adres, kodPocztowy, miasto, telefon, przerwane
- **kategorie** - id\_kategorii, nazwaKategorii
- **szczegolyZamowienia** - id\_zamowienia, id\_produktu, cena, ilosc
- **zamowienia** - id\_zamowienia, id\_klienta, dataZamowienia, dataWysylki, wyslanePrzez, adresWysylki, miastoWysylki, kodPocztowyWysylki, przetworzonePrzez, opis, cena
- **klienci** - id\_klienta, imie, nazwisko, email, telefon, adres, miasto, kodPocztowy
- **logowanie\_klienta** - id\_klienta, login, haslo
- **spedytor** - id\_spedytor, nazwaFirmy, telefon, przerwane
- **pracownicy** - id\_pracownika, imie, nazwisko, pozycja, email, telefon, adres, miasto, kodPocztowy
- **logowanie** - id\_pracownika, login, haslo

## 6. Zaprojektowanie relacji pomiędzy encjami



## III. Projekt logiczny

### 7. Projektowanie tabel, kluczy, indeksów

produkty		
Kolumna	Typ danych	Opis
id_produkty	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
nazwaProduktu	varchar	atrybut reprezentujący nazwę produktu

id_nadawcy	INT	klucz obcy, nawiązujący do tabeli nadawcy
id_kategorii	INT	klucz obcy, nawiązujący do tabeli kategorii
cena	decimal	atrybut reprezentujący cenę produktu
id_magazynu	INT	klucz obcy, nawiązujący do tabeli magazynu
przerwane	boolean	atrybut reprezentujący dostępność produktu

magazyny		
Kolumna	Typ danych	Opis
id_magazynu	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
wlasciciel	varchar	atrybut reprezentujący właściciela magazynu
telefon	varchar	atrybut reprezentujący numer telefoniczny magazynu
email	varchar	atrybut reprezentujący email magazynu
adres	varchar	atrybut reprezentujący adres magazynu
kodPocztowy	varchar	atrybut reprezentujący kodPocztowy magazynu
miasto	varchar	atrybut reprezentujący miasto magazynu
przerwane	boolean	atrybut reprezentujący dostępność magazynu

nadawcy		
Kolumna	Typ danych	Opis
id_nadawcy	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
nazwaFirmy	varchar	atrybut reprezentujący nazwę firmy nadawcy
telefon	varchar	atrybut reprezentujący numer telefoniczny nadawcy
email	varchar	atrybut reprezentujący email nadawcy
adres	varchar	atrybut reprezentujący adres nadawcy
kodPocztowy	varchar	atrybut reprezentujący kod pocztowy nadawcy
miasto	varchar	atrybut reprezentujący miasto nadawcy

przerwane	boolean	atrybut reprezentujący dostępność nadawcy
-----------	---------	---

kategorie		
Kolumna	Typ danych	Opis
id_kategorii	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
nazwaKategorii	varchar	atrybut reprezentujący nazwę kategorii

szczegolyZamowienia		
Kolumna	Typ danych	Opis
id_produktu	INT	klucz główny (obcy), nawiązujący do tabeli produkty
id_zamowienia	varchar	klucz główny (obcy), nawiązujący do tabeli zamowienia
cena	decimal	atrybut reprezentujący cenę produktu przy zamówieniu
ilosc	INT	atrybut reprezentujący ilość zamówionego produktu

zamowienia		
Kolumna	Typ danych	Opis
id_zamowienia	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
id_klienta	INT	klucz obcy, nawiązujący do tabeli klienci
dataZamowienia	date	atrybut reprezentujący datę zamówienia
dataWysylki	date	atrybut reprezentujący datę wysyłki
wyslanePrzez	INT	klucz obcy, nawiązujący do tabeli spedytora
adresWysylki	varchar	atrybut reprezentujący adres wysyłki
miastoWysylki	varchar	atrybut reprezentujący miasto wysyłki

kodPocztowyWysylki	boolean	atrybut reprezentujący kod pocztowy wysyłki
przetworzonePrzez	INT	klucz obcy, nawiązujący do tabeli pracownicy
opis	text	atrybut reprezentujący listę zamówionych produktów oraz ich kategorie
cena	decimal	atrybut reprezentujący końcową cenę zamówienia

klienci		
Kolumna	Typ danych	Opis
id_klienta	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
imie	varchar	atrybut reprezentujący imię klienta
nazwisko	varchar	atrybut reprezentujący nazwisko klienta
telefon	varchar	atrybut reprezentujący numer telefoniczny klienta
email	varchar	atrybut reprezentujący email klienta
adres	varchar	atrybut reprezentujący adres klienta
kodPocztowy	varchar	atrybut reprezentujący kod pocztowy klienta
miasto	varchar	atrybut reprezentujący miasto klienta

logowanie_klienta		
Kolumna	Typ danych	Opis
id_klienta	INT	klucz główny (obcy), nawiązuje do tabeli klienci
login	varchar	atrybut reprezentujący login klienta
haslo	varchar	atrybut reprezentujący hasło klienta

spedytor		
Kolumna	Typ danych	Opis
id_spedytora	INT	klucz główny (obcy), nawiązuje do tabeli klienci
nazwaFirmy	varchar	atrybut reprezentujący nazwę firmy spedytora
telefon	varchar	atrybut reprezentujący numer telefoniczny nadawcy
przerwane	boolean	atrybut reprezentujący dostępność spedytora

pracownicy		
Kolumna	Typ danych	Opis
id_pracownika	INT	klucz główny; jest to automatycznie wygenerowany numer sekwencji
imie	varchar	atrybut reprezentujący imię klienta
nazwisko	varchar	atrybut reprezentujący nazwisko klienta
pozycja	varchar	atrybut reprezentujący pozycję pracownika
telefon	varchar	atrybut reprezentujący numer telefoniczny klienta
email	varchar	atrybut reprezentujący email klienta
adres	varchar	atrybut reprezentujący adres klienta
kodPocztowy	varchar	atrybut reprezentujący kod pocztowy klienta
miasto	varchar	atrybut reprezentujący miasto klienta

logowanie		
Kolumna	Typ danych	Opis
id_pracownika	INT	klucz główny (obcy), nawiązuje do tabeli pracownicy
login	varchar	atrybut reprezentujący login klienta
haslo	varchar	atrybut reprezentujący hasło klienta



## Widoki

informacjaProdukt	informacjaZamowienia
id_produktu	id_produktu
nazwaProduktu	nazwaProduktu
cena	cena_produktu
przerwane	id_zamowienia
id_kategorii	id_klienta
nazwaKategorii	dataWysylki
id_magazynu	dataZamowienia
wlasciciel	wyslanePrzez
telefon_magazynu	adresWysylki
email_magazynu	miastoWysyki
adres_magazynu	kodPocztowyWysylki
kodPocztowy_magazynu	przetworzonePrzez
miasto_magazynu	cena
id_nadawcy	opis
nazwaFirmy	id_kategorii
email_nadawcy	nazwaKategorii
adres_nadawcy	
kodPocztowy_nadawcy	
miasto_nadawcy	
telefon_nadawcy	

**informacjaProdukt** - przechowuje wszystkie informacje powiązane z tabelą produkty (tabele produkty, magazyny, kategorie, nadawcy).

**informacjaZamowienia** - przechowuje wszystkie informacje potrzebne określenia zamówienia(tabele szczegolyZamowienia, zamowienia, produkty, kategorie).

Widoki zostały umieszczone w pliku **views.sql**.

## 8. Zaprojektowanie operacji na danych

- Funkcja **zalogujKlienta** - przy logowaniu jako klient, sprawdza czy podany login oraz hasło są w bazie. Jeżeli tak, to klient loguje się do głównego okna aplikacji. Jeżeli nie, wyrzuca wyjątek.
- Funkcja **zalogujMenadzera** - przy logowaniu jako menedżer, sprawdza czy podany login oraz hasło są w bazie. Jeżeli tak, to menedżer loguje się do głównego okna aplikacji. Jeżeli nie, wyrzuca wyjątek.
- Funkcja **dodajKlienta** - przy tworzeniu konta sprawdza czy istnieje już klient z podanym numerem. Jeżeli nie, to dodaje dane do bazy. Jeżeli tak, to wyrzuca wyjątek.
- Funkcja **dodajProdukt** - dodaje nowy produkt oraz zapisuje kim jest jego nadawca (id nadawcy) oraz w jakim magazynie znajduje się ten produkt.
- Funkcja **dodajMagazyn** - sprawdza czy podany właściciel magazynu już istnieje. Jeżeli nie dodaje nowy magazyn z wprowadzonymi danymi
- Funkcja **dodajNadawce** - sprawdza czy podany nadawca (nazwa firmy) już istnieje. Jeżeli nie dodaje nowego nadawcę z wprowadzonymi danymi
- Wyzwalacz **client\_valid** - uruchamia funkcję **valid\_data** po dodaniu nowego rekordu do tabeli klienci
- Funkcja **valid\_data** - sprawdza czy wprowadzone pola imienia i nazwiska klienta, przy założeniu konta, nie są puste
- Wyzwalacz **category\_valid** - uruchamia funkcję **valid\_category** po dodaniu nowego rekordu do tabeli kategorie
- Funkcja **valid\_data** - sprawdza czy wprowadzone pole z nazwą kategorii nie jest puste
- Wyzwalacz **login\_worker\_valid** - uruchamia funkcję **valid\_login\_worker** po dodaniu nowego rekordu do tabeli pracownicy
- Funkcja **valid\_login\_worker** - sprawdza czy wprowadzone pola z loginem oraz hasłem po zalogowaniu jako menedżer nie są puste
- Wyzwalacz **login\_client\_valid** - uruchamia funkcję **valid\_login\_client** po dodaniu nowego rekordu do tabeli pracownicy
- Funkcja **valid\_login\_client** - sprawdza czy wprowadzone pola z loginem oraz hasłem po zalogowaniu jako klient nie są puste

Kod wszystkich funkcji i wyzwalaczy został umieszczony w plikach **functions.sql** oraz **triggers.sql** odpowiednio.

## IV. Projekt funkcjonalny

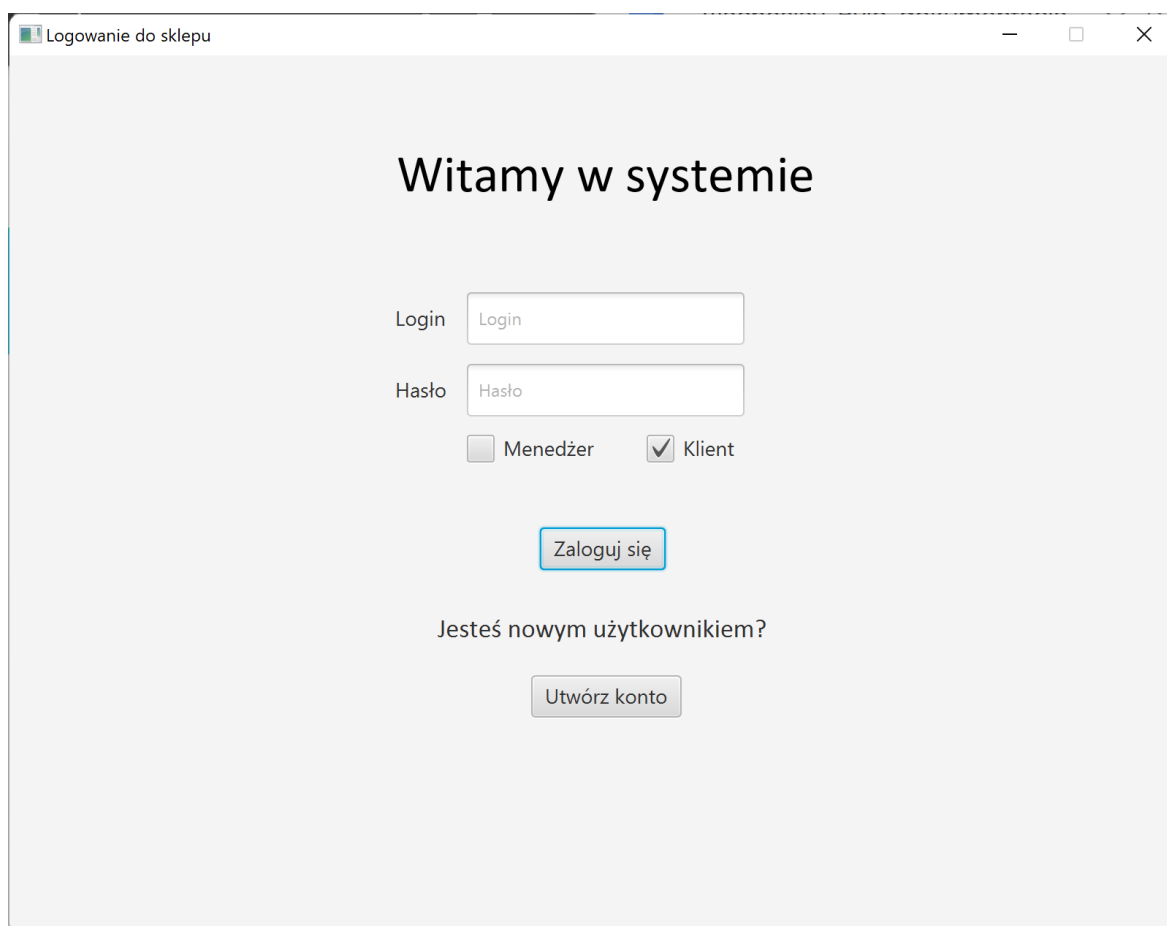
### 9. Interfejsy do prezentacji, edycji i obsługi danych, wizualizacja danych, zdefiniowanie panelu sterowania aplikacji

#### Technologia JavaFX.

Wszystkie dane są wizualizowane za pomocą tabel (TableView JavaFX) oraz „boksów” (ComboBox JavaFX). Także są komunikaty (Alert JavaFX) informujące o potwierdzeniu dodania rekordu do bazy, a także w przypadku jakiegokolwiek błędu - wyświetlają wskazówki co do poprawności danych.

Testowe dane dla klienta: numer telefonu: login: karol, hasło: karol

Testowe dane dla dyrektora: login: asd, hasło: 123.



Rysunek 1. Okno logowania aplikacji. 1. – pola “Login” oraz “Hasło” do wprowadzania danych, 2. – wybór roli, 3. – przyciski dla logowania oraz tworzenia konta.

W przypadku gdy podane dane są nieprawidłowe aplikacja wyświetla komunikat, w przeciwnym przypadku przekieruje użytkownika do głównego okna aplikacji.

Sklep internetowy

Powrót

## Rejestracja konta

### Informacje osobiste

Imię

Nazwisko

Email

Numer telefonu

Adres

Miasto

Kod pocztowy

### Informacje o koncie

Login

Hasło

Utwórz

Rysunek 2. Okno aplikacji gdzie klient może utworzyć swoje konto

Po utworzeniu konta, aplikacja wyświetla komunikat o tym że konto zostało utworzone i przekierowuje użytkownika do okna logowania. Po wprowadzeniu nietypowych danych (błąd w imieniu, nazwisku, numerze telefonu itd.) – wyświetla komunikat, że jest jakiś błąd z komunikatem w którym miejscu jest błąd i prosi użytkownika jeszcze raz sprawdzić swoje dane.

Sklep internetowy

Elektronika

Wyloguj się

Wyszukaj

Produkty	Cena
telefon	100.0

Produkty	Cena	Ilość
telefon	100.0	1

**Adres do wysyłki:**

Ulica:

Miasto:

Kod pocztowy:

Zmień adres wysyłki

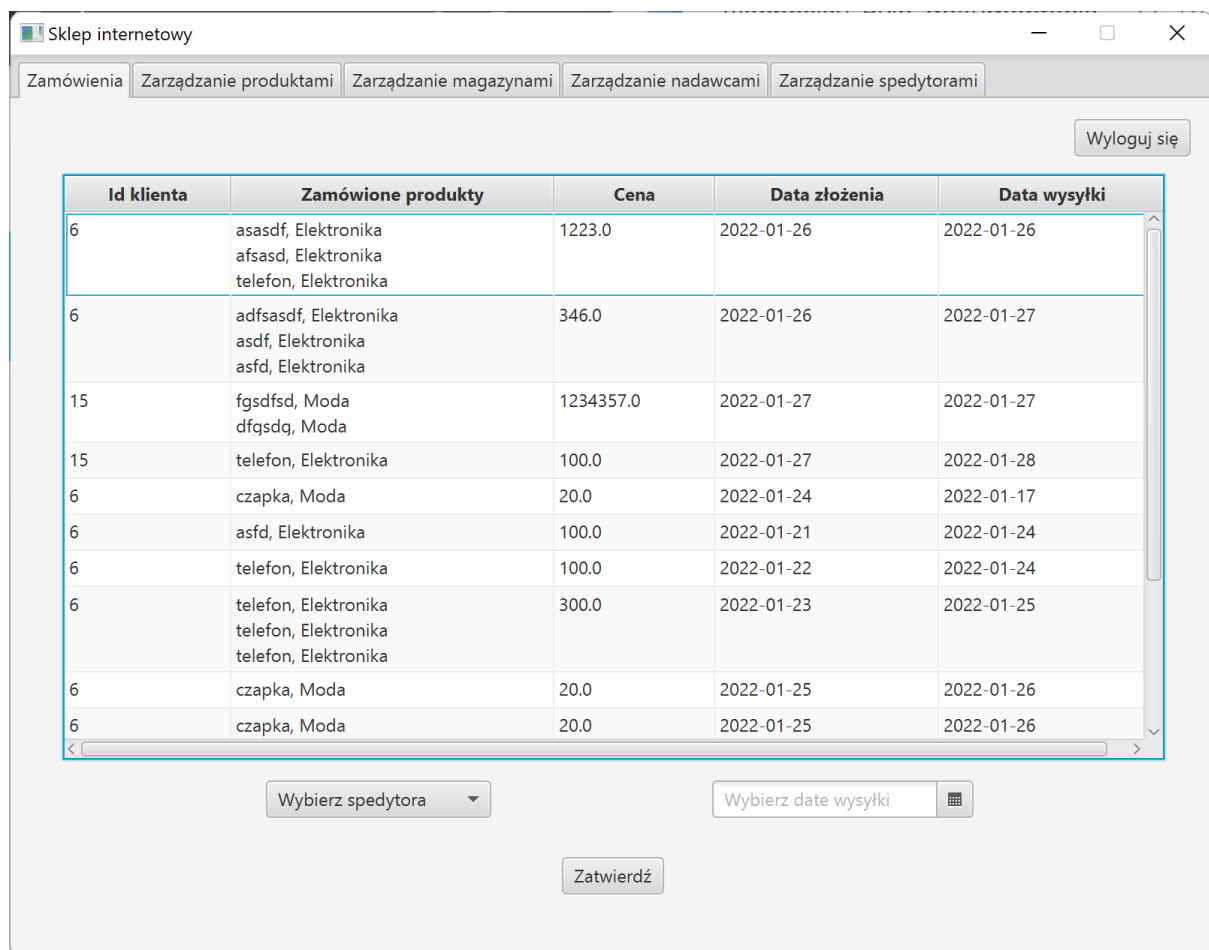
Dodaj

Usuń

Kupić

Rysunek 3. Okno aplikacji po zalogowaniu jako klient.

W danym oknie klient może wybrać jedną z kategorii dostępnych na liście. W zależności od wybranej kategorii aplikacja wylistuje tabele dostępnych produktów. Dalej klient ma możliwość wybrać produkt. Jeżeli klient chce zmienić swój adres wysyłki na inny niż był podany przy tworzeniu konta, to może to zrobić w polu “Adres do wysyłki”.

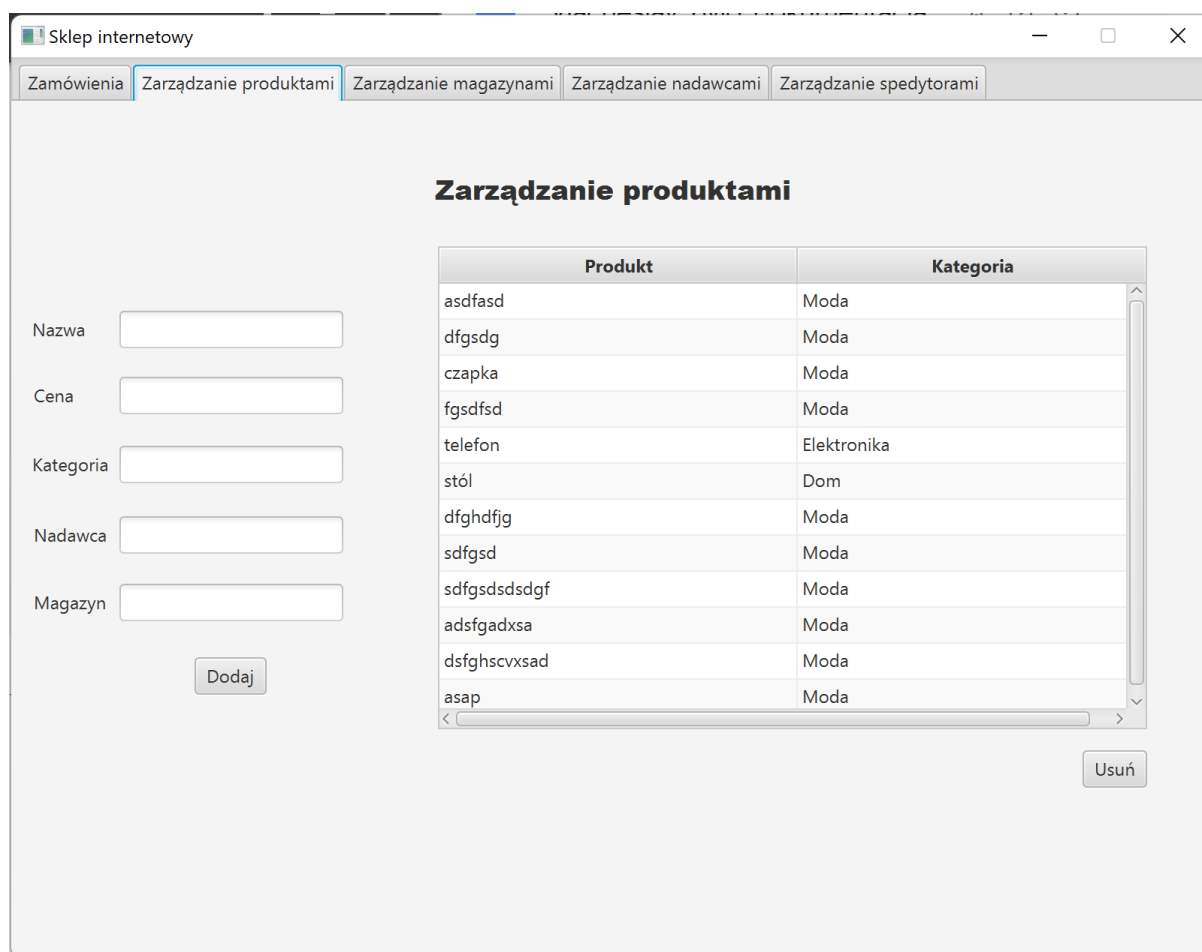


Rysunek 4. Okno aplikacji po zalogowaniu jako menedżer.

Po zalogowaniu menedżer może wybrać jedną z 5 opcji:

- Zamówienia
- Zarządzanie produktami
- Zarządzanie magazynami
- Zarządzanie nadawcami
- Zarządzanie spedytorami

Standardowo, pierwszą wybraną zakładką jest "Zamówienia". Tutaj menedżer może wybrać spedytora z listy oraz datę wysyłki, i wtedy wysłać wybrany z tabeli produkt.



Rysunek 5. Okno aplikacji po naciśnięciu na patkę "Zarządzanie produktami".

1. – W polach "Nazwa", "Cena", "Kategoria", "Nadawca", "Magazyn" można wpisać dane dla nowego produktu.
2. – przycisk "Dodaj" doda nowy produkt do tabeli.
3. – przycisk "Usuń" usunie wybrany produkt z tabeli.

W wybranej patce menedżer może dodać nowy produkt, gdzie w polach "Kategoria", "Nadawca" i "Magazyn" wpisze już istniejące nazwy.

Sklep internetowy

Zamówienia

Zarządzanie produktami

Zarządzanie magazynami

Zarządzanie nadawcami

Zarządzanie spedytorami

Zarządzanie magazynami

Właściciel

Email

Ulica

Kod pocztowy

Miasto

Telefon

Dodaj magazyn

Właściciel	Miasto	telefon	Email	Ilość produktów
afasda	asdf	123456788	asdf@mail.c...	2
asdf	adsf	123456789	asd@fdas.c...	4
sdf	sdfa	123456789	asdf@asdf.c...	4
asfdasd	asdf	123456780	asdf@gma.c...	
asdfas	asdf	123456788	asdf@asdf.c...	
asfdasd	asdf	123456788	asdf@asdf.c...	

Usuń magazyn

Rysunek 6. Okno aplikacji po naciśnięciu na patkę “Zarządzanie magazynami”.

1. – W polach “Właściciel”, “Email”, “Ulica”, “Kod pocztowy”, “Miasto”, “Telefon” można wpisać dane dla nowego magazynu.
2. – przycisk “Dodaj magazyn” doda nowy magazyn do tabeli.
3. – przycisk “Usuń magazyn ” usunie wybrany magazyn z tabeli.

W wybranej patce menedżer może dodać nowy magazyn, gdzie musi podać prawidłowe dane, inaczej wyskoczy błąd o niepoprawności danych.



Sklep internetowy

Zamówienia

Zarządzanie produktami

Zarządzanie magazynami

Zarządzanie nadawcami

Zarządzanie spedytorami

### Zarządzanie nadawcami

Nazwa firmy

Email

Ulica

Kod pocztowy

Miasto

Telefon

Dodaj nadawcę

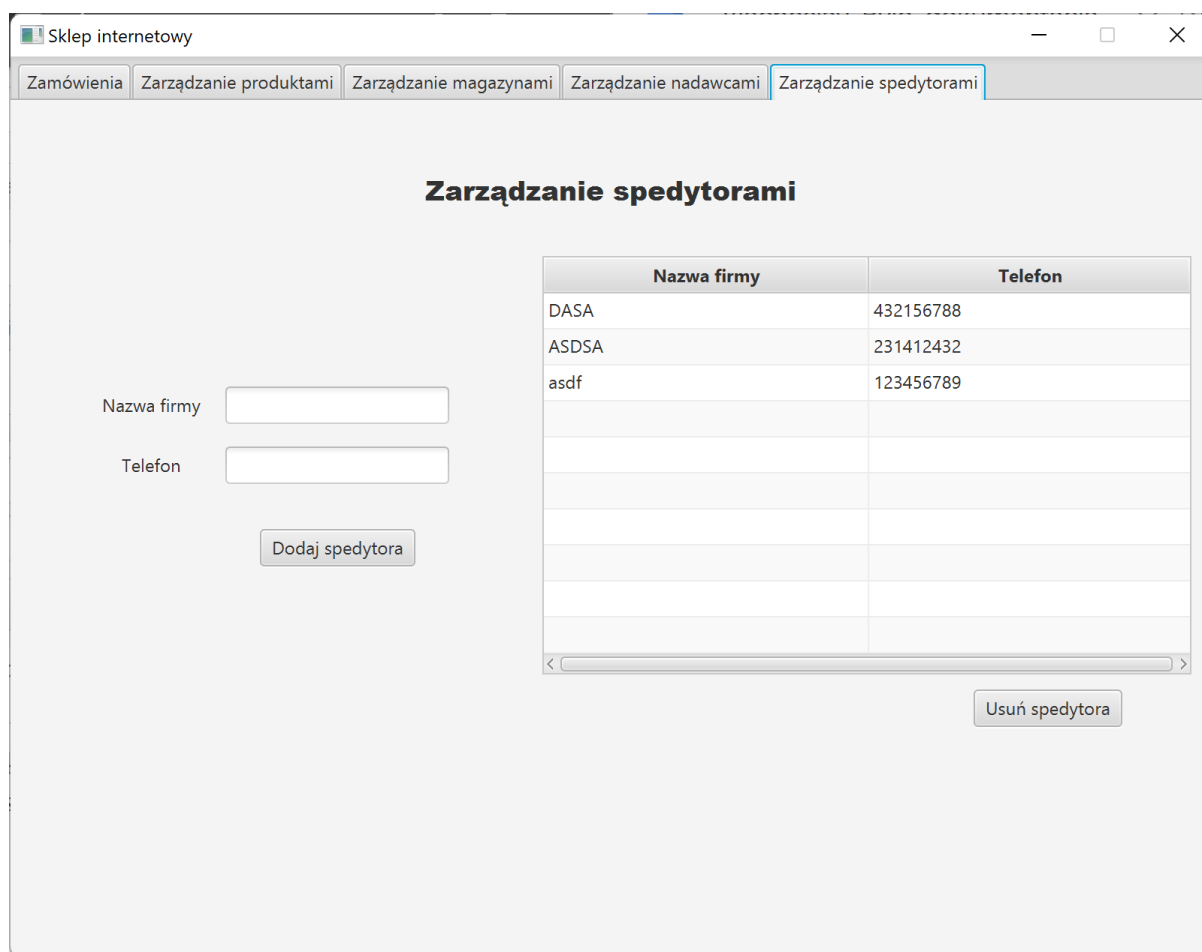
Nazwa firmy	Ulica	Miasto	Email	Telefon
ALPACH	Reumonta 17	Warszawa	alpach@e...	123456781
AHGSD	ASdas	safasd	hasd@mail...	142134123
asdgsaf	asdf	asdfa	asdf@asdf...	123456789
safas	asdfasdf	asfasdf	asdf@mail...	123456789

Usuń nadawcę

Rysunek 7. Okno aplikacji po naciśnięciu na patkę “Zarządzanie nadawcami”.

1. – W polach “Nazwa firmy”, “Email”, “Ulica”, “Kod pocztowy”, “Miasto”, “Telefon” można wpisać dane dla nowego nadawcy.
2. – przycisk “Dodaj nadawcę” doda nowego nadawcę do tabeli.
3. – przycisk “Usuń nadawcę” usunie wybranego nadawcę z tabeli.

W wybranej patce menedżer może dodać nowego nadawcę, gdzie musi podać prawidłowe dane, inaczej wyskoczy błąd o niepoprawności danych.



Rysunek 7. Okno aplikacji po naciśnięciu na patkę „Zarządzanie spedytorami”.

1. – W polach „Nazwa firmy”, „Telefon” można wpisać dane dla nowego spedytora.
2. – przycisk „Dodaj spedytora” doda nowego spedytora do tabeli.
3. – przycisk „Usuń spedytora” usunie wybranego spedytora z tabeli.

W wybranej patce menedżer może dodać nowego nadawcę, gdzie musi podać prawidłowe dane, inaczej wyskoczy błąd o niepoprawności danych.

## 10. Makropolecenia

Aby nie wprowadzać wszystkich informacji ręcznie, w miarę możliwości aplikacja posiada „ComboBox” oraz „TableView”, co jest ułatwieniem dla użytkownika.

## V. Dokumentacja

### 11. Wprowadzanie danych: zdefiniowanie sposobu wprowadzania danych

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

Ręcznie wprowadzone zostały tylko dane dla menedżera. Resztę danych można wprowadzić za pomocą aplikacji (automatycznie).

## 12. Dokumentacja użytkownika

Folder **“Viacheslvav\_Bylo\_projekt”** należy wyciągnąć z archiwum w dowolne miejsce na komputerze. Dalej w tym folderze uruchomić plik **SklepInternetowy.jar**.  
**Uwaga – dla uruchomienia pliku SklepInternetowy.jar jest potrzebny JDK wersji 17.** Link do ściągnięcia tej wersji jest podany niżej. Baza danych stworzona jest w PostgreSQL. Wszystkie potrzebne tabele, funkcje i widoki zostały już dodane. Baza danych jest umieszczona na darmowym hostingu ElephantSQL.

[https://download.oracle.com/java/17/archive/jdk-17.0.2\\_windows-x64\\_bin.exe](https://download.oracle.com/java/17/archive/jdk-17.0.2_windows-x64_bin.exe)

## 13. Opracowanie dokumentacji technicznej

Kod został udokumentowany w kodzie źródłowym.

## 14. Wykaz literatury

Przy tworzeniu projektu korzystano z:

- wykładów do przedmiotu Bazy Danych I profesora Antoniego Dydejczyka - [https://newton.fis.agh.edu.pl/~antek/index.php?sub=db\\_wykl](https://newton.fis.agh.edu.pl/~antek/index.php?sub=db_wykl)
- materiałów z laboratoriów z Andrzejem Lemańskim
- Dokumentacja JavaFx - [JavaFX](#)
- Dokumentacja PostgreSQL - [PostgreSQL: Documentation](#)