Dokumentacja projektu Bazy Danych 1

Jędrzej Szostak

Link do aplikacji - http://pascal.fis.agh.edu.pl/~9szostak/Projekt2/index.php

I. Projekt koncepcji, założenia

1. *Zdefiniowanie tematu projektu:*

Tematem projektu jest obsługa sieci sklepów spożywczych (np. takich jak Żabka). Ma pozwalać pracownikom szeregowym zarządzać konkretnymi jednostkami do których są przypisani, w tym dodawanie zamówień, zarządzanie nimi, podgląd stanu magazynowego sklepu i uzupełnianie go. Z kolei kierownik zarządza pracownikami (zatrudnia, zwalnia, edytuje ich dane) oraz wszystkimi sklepami. Osoby na obu stanowiskach mogą też przeglądać swoje informacje personalne i zmieniac haslo dostepu.

2. *Analiza wymagań użytkownika:*

Funkcjonalności zostały pokrótce przedstawione powyżej. Aplikacja ma dwie warstwy, jedna dla pracownika szeregowego, a druga dla kierownika. Obie warstwy wymagają logowania, aby się do nich dostać.

Kierownik:
Ma on możliwość przeglądania informacji personalnych, zmianę hasła dostępu, przeglądanie wszystkich pracowników, edytowanie ich informacji personalnych, dodawanie wraz z przypisywaniem ich do konkretnych sklepów i zwalnianie pracowników. Przegląd sklepów oraz dodawanie nowych sklepów (jeśli firma się rozrasta :)).

Sprzedawca:
Ma on tak jak kierownik mozliwosc przegladania swoich informacji personalnych oraz, zmianę hasła dostępu. Może również przeglądać listę produktów dostępnych w sklepach i dodawać jakieś nowe. Ma możliwość również przeglądać stan i ilość produktów w jego sklepie oraz uzupełniać ich stan. Ma również dostęp do wszystkich sprzedanych produktów w jego sklepie. Jako sprzedawca ma również dostęp oczywiście do złożonych zamówień, które sam dodaje wraz z informacjami o klientach, może je również przeglądać i zmieniać status zamówienia na zakończone jeśli dostał wpłatę do konkretnego zamówienia.

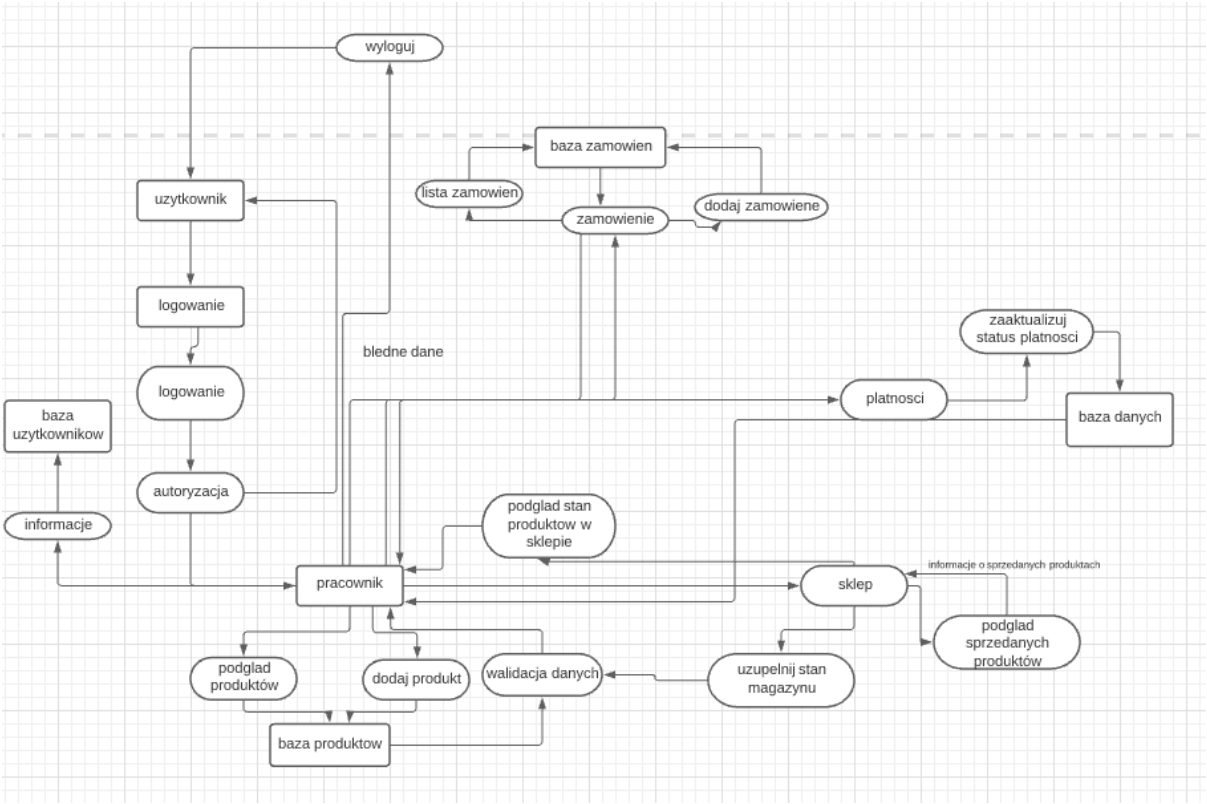
3. *Zaprojektowanie funkcji:*

Podstawowe funkcje realizują funkcjonalności opisane powyżej. Jej główna funkcja jest przechowywanie ifnormacji o pracownikach, sklepach, produktach, producentach tych produktów, klientach, zamówieniach, płatnościach i działach w sklepach.

II. Projekt diagramów (konceptualny)

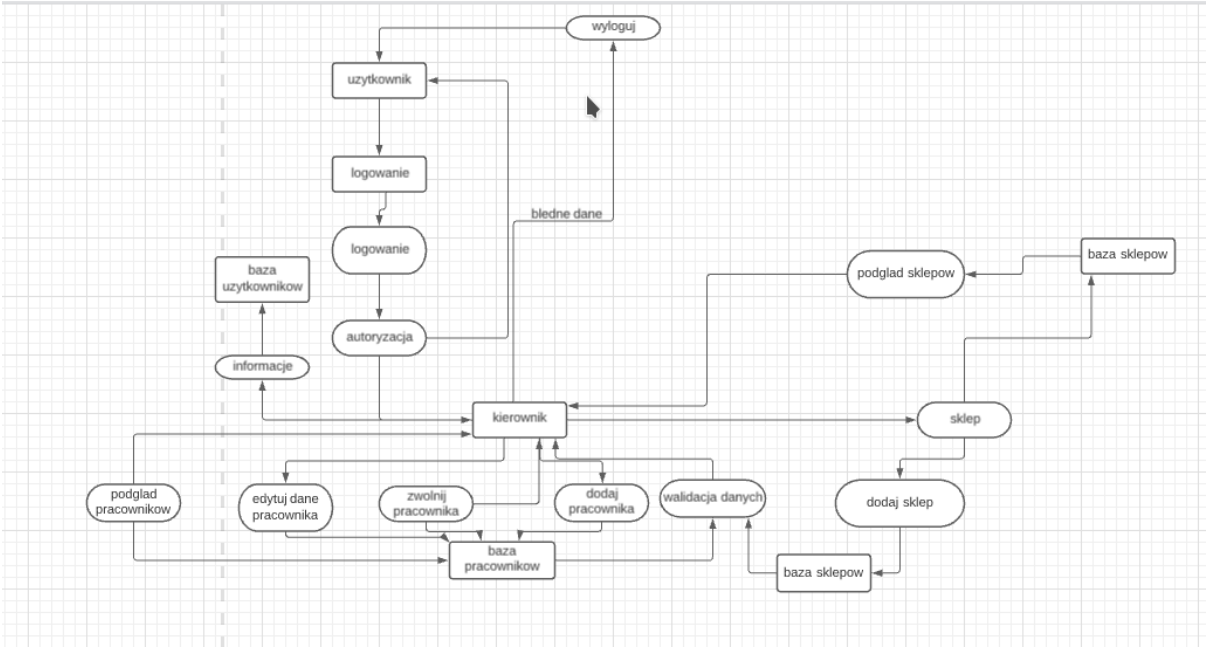
4. *Budowa i analiza diagramu przepływu danych (DFD):*

DFD dla sprzedawcy:



Rys1. Diagram DFD dla sprzedawcy.

DFD dla kierownika:



Rys2. Diagram DFD dla kierownika.

5. *Zdefiniowanie encji (obiektów) oraz ich atrybutów:*

dzial: id_dzial, opis, nazwa

produkt: id_produkt, id_dzial, nazwa, cena, id_pracownik

producent: id_producent, nazwa, email, telefon

magazyn_produkt: id_produkt, id_magazyn, ilosc

zamowienie_szczegoly: id_produkt, id_magazyn, id_zamowienie, ilosc

zamowienie: id_zamowienie, id_klient, status, data, kwota

metoda_platnosci: id_metoda_platnosci, id_zamowienie, metoda_platnosci, data_platnosci

klient: id_klient, id_adres, nazwisko, imie, email, nip

adres: id_adres, miasto, ulica_i_numer, kod_pocztowy

magazyn: id_magazyn, id_adres, glowny

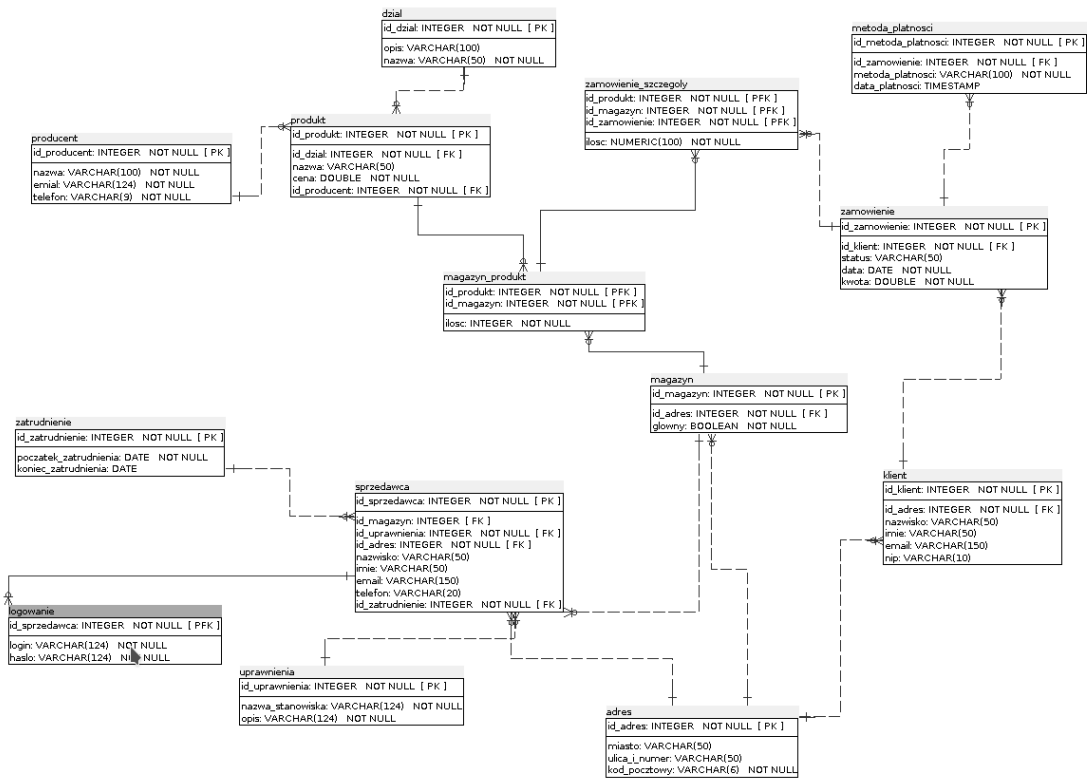
sprzedawca: id_sprzedawca, id_magazyn, id_uprawnienia, id_adres, nazwisko, imie, email, telefon, id_zatrudnienie

uprawnienia: id_uprawnienia, nazwa_stanowiska, opis

logowanie: id_sprzedawca, login, haslo

zatrudnienie: id_zatrudnienie, poczatek_zatrudnienia, koniec_zatrudnienia

6. *Zaprojektowanie relacji między encjami:*



Rys3. Diagram ERD.

III. Projekt logiczny

7. Projektowanie tabel, kluczy, indeksów:

Cały skrypt pozwalający na stworzenie całej struktury na podstawie diagramu ERD powyżej znajduje się w pliku ktory zostal przeslany w katalogu base/insert.sql.

8. Słowniki danych:

Schemat proj:

- tabela producent:
  - id_producent INTEGER NOT NULL PRIMARY KEY
  - nazwa VARCHAR (100) NOT NULL
  - email VARCHAR (124) NOT NULL
  - telefon VARCHAR (9) NOT NULL
- tabela produkt:
  - id_produkt INTEGER NOT NULL PRIMARY KEY
  - id_dzial INTEGER NOT NULL (FK) - klucz obcy do tabeli dzial
  - nazwa VARCHAR (50) NOT NULL
  - cena DOUBLE NOT NULL
  - id_producent INTEGER NOT NULL (FK) - klucz obcy do tabeli producent

- tabela dzial
    - id_dzial INTEGER NOT NULL PRIMARY KEY
    - opIs VARCHAR (100)
    - nazwa VARCHAR (100)
- tabela magazyn_produkt:
    - id_produkt INTEGER NOT NULL (PFK) - klucz do tabeli produkt
    - id_magazyn INTEGER NOT NULL (PFK) - klucz do tabeli magazyn
    - ilosc INTEGER NOT NULL
- tabela zamowienie_szczegoly
    - id_produkt: INTEGER NOT NULL (PFK) - klucz do tabeli produkt
    - id_magazyn: INTEGER NOT NULL (PFK) - klucz do tabeli magazyn
    - id_zamowienie: INTEGER NOT NULL (PFK) - klucz fo tabeli zamowienie
    - ilosc INTEGER NOT NULL
- tabela zamowienie
    - id_zamowienie: INTEGER NOT NULL (PK)
    - id_klient: INTEGER NOT NULL (FK) - klucz do tabeli klient
    - status: VARCHAR (50)
    - data: DATE NOT NULL
    - kwota: DECIMAL NOT NULL
- tabela metoda_platnosci
    - id_metoda_platnosc:i INTEGER NOT NULL (PK)
    - id_zamowienie: INTEGER NOT NULL (FK) - klucz to tabeli zamowienie
    - metoda_platnosci: VARCHAR (100) NOT NULL
    - data_platnosci: TIMESTAMP
- tabela klient
    - id_klient: INTEGER NOT NULL (PK)
    - id_adres: INTEGER NOT NULL (FK) - klucz to tabeli adres
    - nazwisko: VARCHAR (50)
    - imie: VARCHAR (50)
    - email: VARCHAR (50)
    - nip: VARCHAR (10)
- tabela adres
    - id_adres: INTEGER NOT NULL (PK)
    - miasto: VARCHAR (50)
    - ulica_i_numer: VARCHAR (50)
    - kod_pocztowy: VARCHAR (6) NOT NULL
- tabela sprzedawca
    - id_sprzedawca: INTEGER NOT NULL (PK)
    - id_magazyn: INTEGER (FK) - klucz do tabeli magazyn
    - id_uprawnienia: INTEGER NOT NULL (FK) - klucz do tabeli uprawnienia
    - id_adres: INTEGER NOT NULL (FK) - klucz to tabeli adres
    - nazwisko: VARCHAR (50)
    - imie: VARCHAR (50)
    - email: VARCHAR (150)

        - telefon: VARCHAR (9)
        - id_zatrudnienie: INTEGER NOT NULL (FK) - klucz do tabeli zatrudnienie
    - tabela uprawnienia
        - id_uprawnienia: INTEGER NOT NULL (PK)
        - nazwa_stanowiska: VARCHAR (124) NOT NULL
        - opis: VARCHAR (124) NOT NULL
    - tabela logowanie
        - id_sprzedawca: INTEGER NOT NULL (PFK) - klucz do tabeli sprzedawca
        - login: VARCHAR (124) NOT NULL
        - haslo: VARCHAR (124) NOT NULL
    - tabela zatrudnienie:
        - id_zatrudnienie: INTEGER NOT NULL (PK)
        - poczatek_zatrudnienia: DATE NOT NULL
        - koniec_zatrudnienia: DATE;
    - tabela magazyn:
        - id_magazyn: INTEGER NOT NULL (PK)
        - id_adres: INTEGER NOT NULL (FK) - klucz do tabeli adres
        - glowny: BOOLEAN NOT NULL

9. *Analiza zależności funkcyjnych i normalizacja tabel:*

   1NF:

   W każdej tabeli jest jednoznacznie definiujący encje klucz główny. Wszystkie dane są atomowe.

   2NF:

   Każda tabela przechowuje dane dotyczące tylko konkretnej klasy obiektów. Każdy niekluczowy atrybut jest nieredukowalnie zależny od klucza głównego.

   3NF:

   Wszystkie niekluczowe atrybuty są wzajemnie niezależne oraz nieredukowalnie zależne od klucza głównego.

   Zoptymalizowanie tabel do 3NF zostało zapewnione na etapie projektowania bazy danych.

11. *Zaprojektowanie operacji na danych:*

    Funkcje:
    - proj.dodajAdres(ul_i_nr VARCHAR, kp VARCHAR(6), m VARCHAR) - funkcja insertujaca do tabeli adres adres o podanych danych, ulicy, kodu pocztowego

i miasta, zwraca wartość true jeśli operacja się uda i false jeśli nie

- proj.dodajMagazyn(main BOOLEAN, ul_i_numer VARCHAR , kp VARCHAR (6), m VARCHAR) - funkcja dodająca sklep do tabeli magazyn, wywołuje wewnątrz funkcje dodaj adres zeby nadac magazynowi adres i sprawdza czy jest to magazyn główny, zwraca true jeśli operacja się powiedzie, false jeśli nie.
- proj.dodajPracownika(stanowisko VARCHAR, dataStartu DATE, id_magazynu INTEGER, imie VARCHAR, nazwisko VARCHAR, email VARCHAR, telefon VARCHAR, ul_i_nr VARCHAR, kp VARVHAR (6), m VARCHAR, l VARCHAR, h VARCHAR) - metoda realizująca dodanie nowego pracownika przez kierownika do bazy pracowników, ustawiane są wszystkie dane pracownika oraz jego login i hasło, wywołuje też wewnątrz funkcje dodajAdres, aby nadać pracownikowi adres. Zwraca true jeśli się powiedzie, false jeśli nie.
- proj.zwolnijMagazyniera(id_sprzed INTEGER) - funkcja realizująca zwolnienie pracownika przez menadżera, dodawana jest data_zwolnienia, która określa że pracownik już nie pracuje. Zwraca true jeśli operacja się powiedzie, false jeśli nie.
- proj.edytujDanePracownika(id_sprzed INTEGER, im VARCHAR, naz VARCHAR, mail VARCHAR, tel VARCHAR, log VARCHAR) - funkcja pozwalająca kierownikowi zmienić dane pracowników oprócz adresu zamieszkania i hasła (login tez mozna zmienic). Zwraca true jeśli się powiedzie, false jeśli nie.
- proj.dodajProdukt(id_kat INTEGER, id_prod INTEGER, n VARCHAR, cena FLOAT) - funkcja pozwalająca pracownikowi dodać produkt do oferty sklepów, Zwraca true jeśli się powiedzie, false jeśli nie.
- proj.dodajAsortyment(id_sk INTEGER, id_prod INTEGER, il INTEGER) - funkcja realizująca dodanie przez pracownika produktów w danej ilości na stan sklepu (uzupełnienie asortymentu). Zwraca true jeśli się powiedzie, false jeśli nie.
- proj.inicjacja_produktu() - funkcja zwracająca trigger ustawiająca każdemu dodanemu produktowi ilość na stanie 0 na początek.
- proj.dodajKlienta(im VARCHAR, naz VARCHAR, email, VARCHAR, nip VARCHAR (10), ul_i_nr VARCHAR (6), kp VARCHAR, m VARCHAR) - funkcja realizująca dodanie klienta do bazy danych, aby dodać jego adres wywoluje funkcje proj.dodajAdres. Zwraca true jeśli się powiedzie, false jeśli nie.
- proj.noweZamowienie(im VARCHAR, naz VARCHAR, email VARCHAR, NIP VARCHAR(10), ul_i_nr VARCHAR, , VARCHAR) - funkcja tworząca nowe zamówienie, najpierw dodaje klienta do bazy danych (wywołując funkcje proj.dodajKlienta). Zwraca id nowo utworzonego zamówienia lub NULL jeśli operacja się nie powiedzie.
- proj.dodajDoZamowienie (id_zam INTEGER, id_prod INTEGER, id_mag INTEGER, il INTEGER) - oblicza wartość zamówienia i dodaje je do konkretnego zamówienia. Zwraca true jeśli się powiedzie, false jeśli nie.
- proj,zaktualizujPlatnosc(id_plat INTEGER, id_zam INTEGER, id_kli INTEGER, ns VARCHAR(12)) - funkcja pozwalająca pracownikowi zmienić

status płatności na zakończone po otrzymaniu wpłaty. Zwraca true jeśli się powiedzie, false jeśli nie.

TRIGGER:

- proj.inicjuj_produkt_magazyn_stan - trigger ustawiający stan nowo dodanego produktu w magazynie na 0, działa dla funkcji inicjacja_produktu().

WIDOKI:

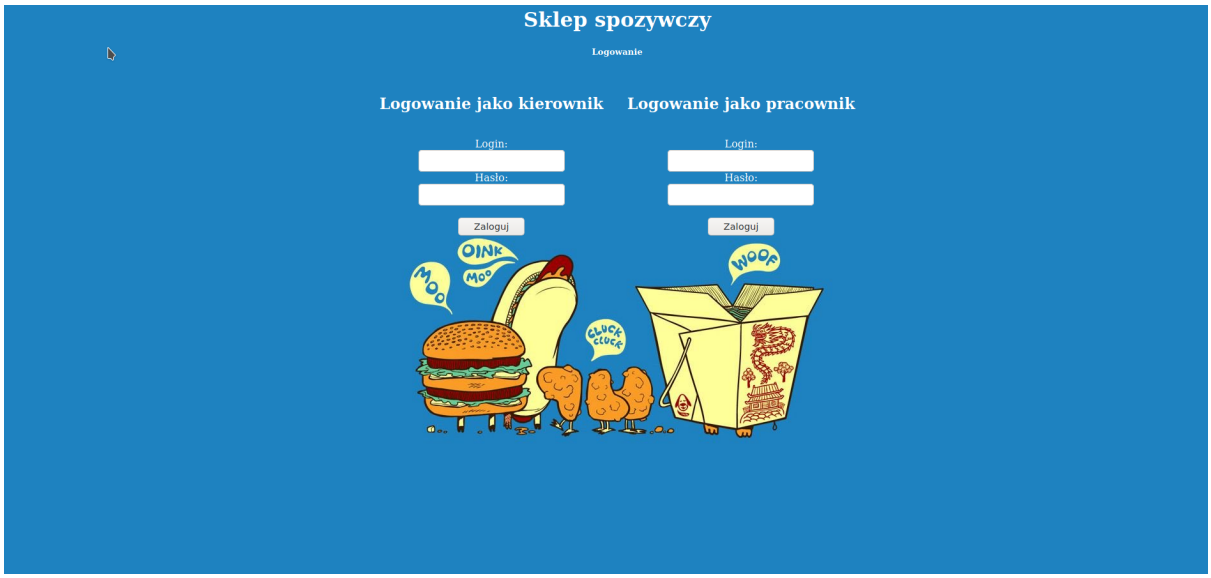
- proj.pracownicy_lista - tworzy widok pracowników ze wszystkimi informacjami
- proj.magazyn_lista - tworzy widok sklepu, z informacjami, adresem i pracownikami odpowiedzialnymi za sklep.
- proj.pracownicy_zatrudnienie_okres - widok pracowników wraz z okresem ich zatrudnienia.
- proj.produkty_lista - tworzenie informacji o produktach wraz z działem i producentem.
- proj.pelny_stan_magazynow - widok sklepu wraz z ich asortymentem.
- proj.pelny_widok_szczegoly_zamowienia - pelny widok szczegolow zamowienia wraz z informacjami o produkcie, sklepie i ilości.
- proj.zamowienia_lista - zamówienia z pełnymi informacjami.
- proj.platnosci_lista - zamówienia wraz z płatnościami .

IV. Projekt funkcjonalny

W stronie domowej mamy interfejs logowania. Do wyboru są dwie opcje:

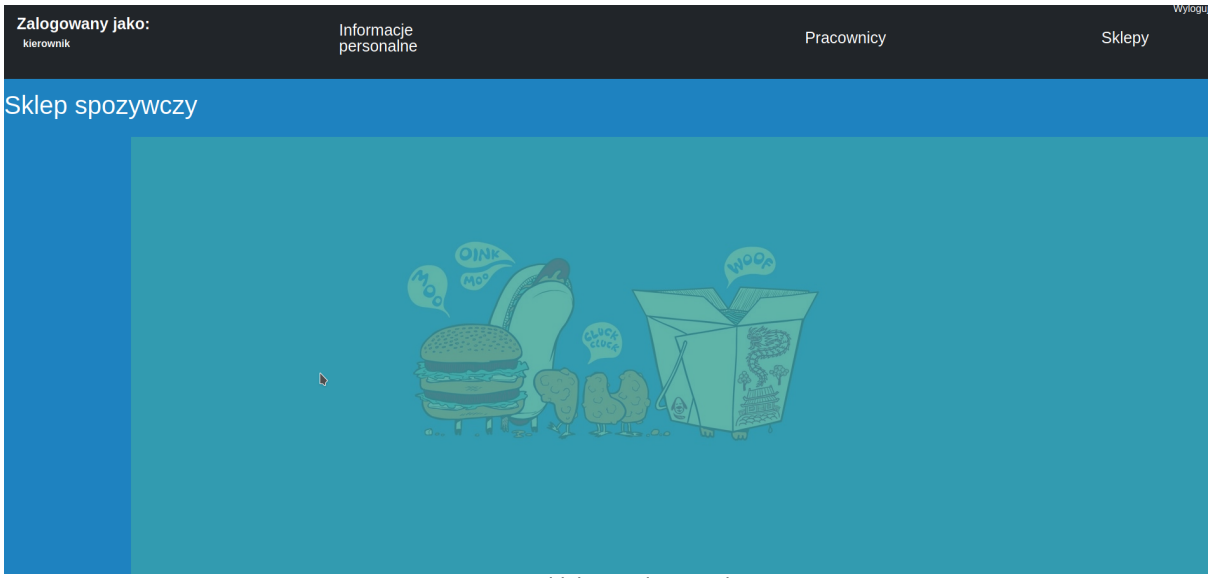
- logowanie jako pracownik
- logowanie jako kierownik

Dokonywana jest walidacja czy wprowadzone dane są prawidłowe i czy osoba o podanych danych znajduje się w bazie. Jeśli tak to następuje przekierowanie i zalogowanie, a jeśli nie to wyskakuje komunikat o błędnych danych logowania (zwolniony pracownik również nie ma dostępu do serwisu).



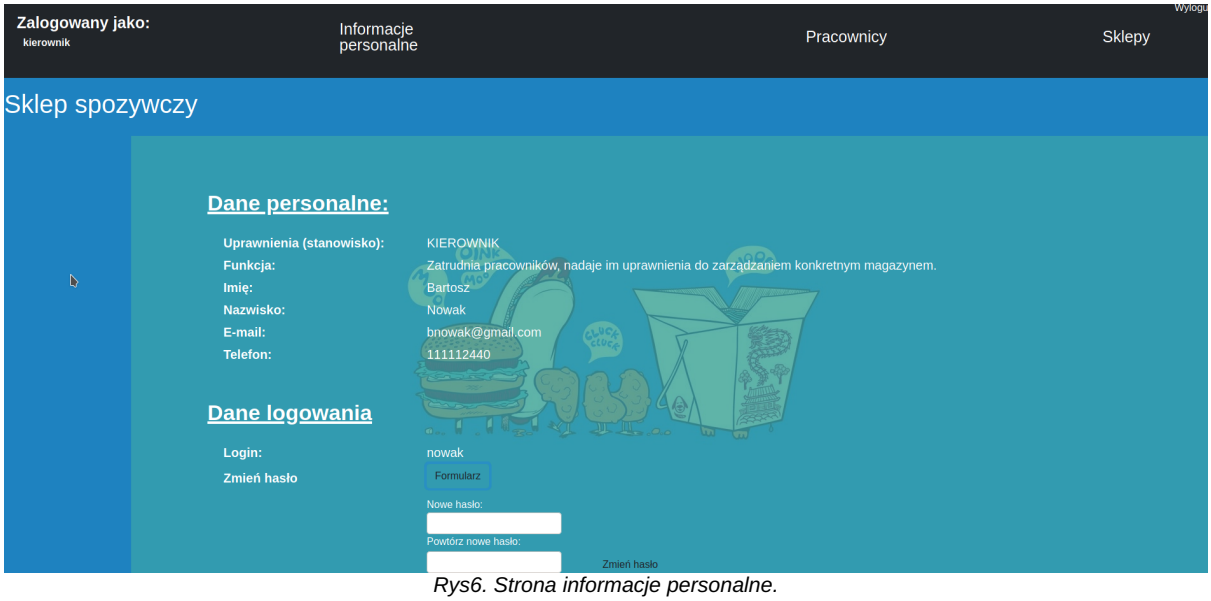
Rys4. Ekran logowania

Po zalogowaniu użytkownik widzi strone domowa, na gorze są dostępne opcje oraz po lewej stronie informacja jako kto został zalogowany.



Rys5. Widok po zalogowaniu.

Zakładka informacje personalne jest identyczna dla obu rol, użytkownik może tam znaleźć swoje dane oraz zmienić, dane do logowania, co jest realizowane za pomocą prostego formularza.



Rys6. Strona informacje personalne.

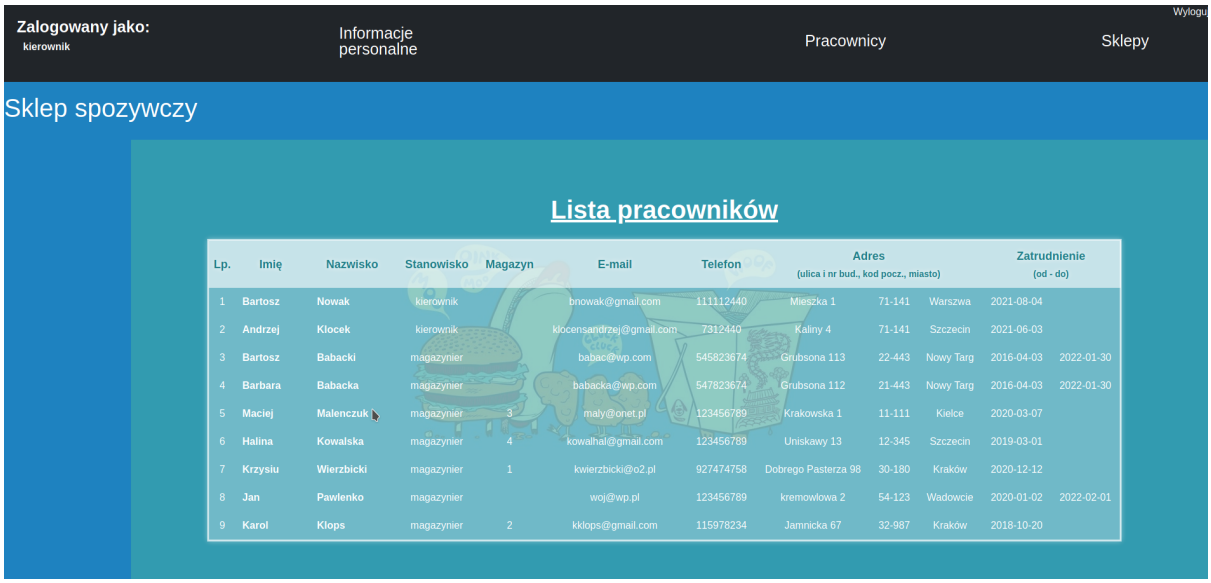
Jako iż kierownik i magazynier mają dostęp do różnych funkcji to opis zostanie podzielony.

1. *Kierownik*

   Pracownicy:

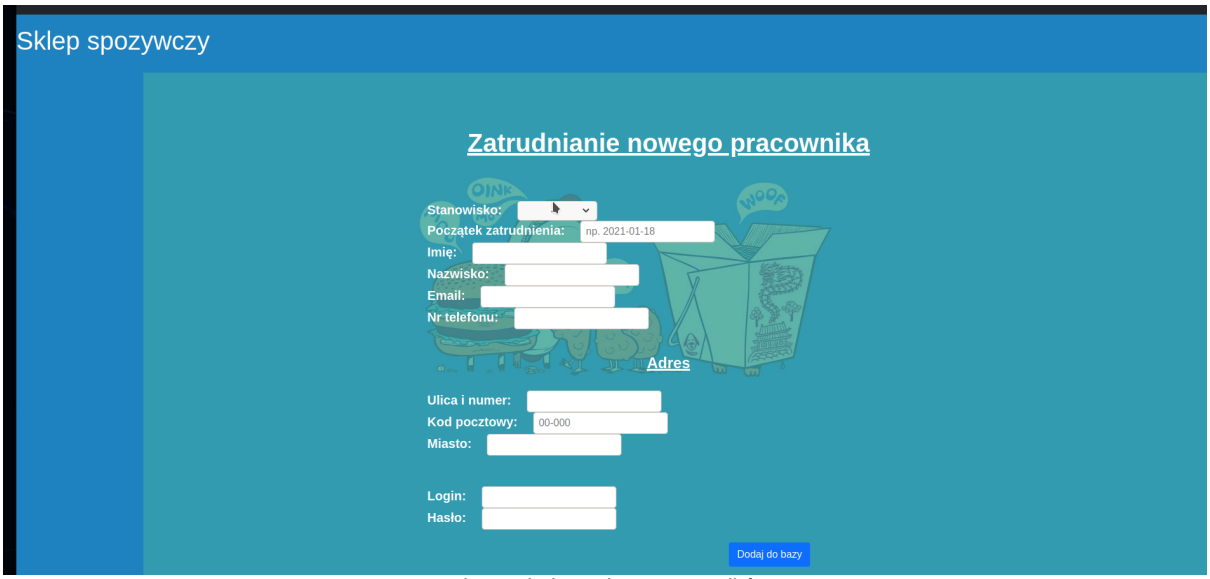
   Zakładka pracownicy zawiera trzy opcje do wyboru: Lista, dodaj, edytuj i zwolnij.

   Lista - jest przedstawiona jako tabela, w której znajdują się podstawowe informacje o pracownikach jak i przydział do sklepu oraz okres zatrudnienia.



Rys7. Lista pracowników.

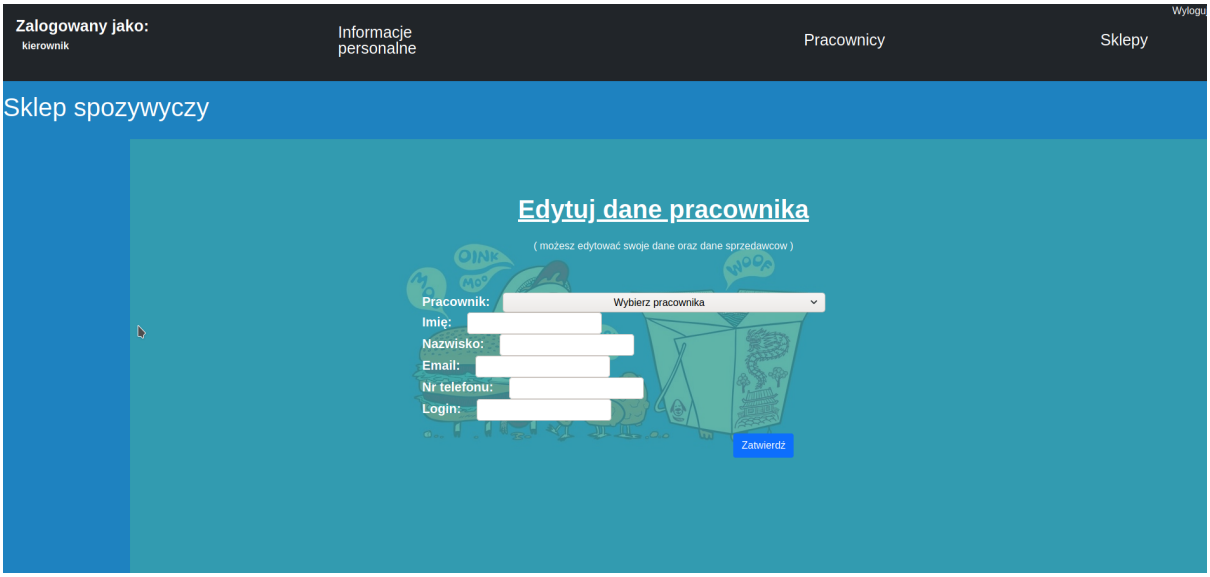
Dodaj - pozwala na zatrudnienie nowego pracownika po uzupełnieniu poniższego formularza:



Rys8. Formularz z dodawaniem pracowników.

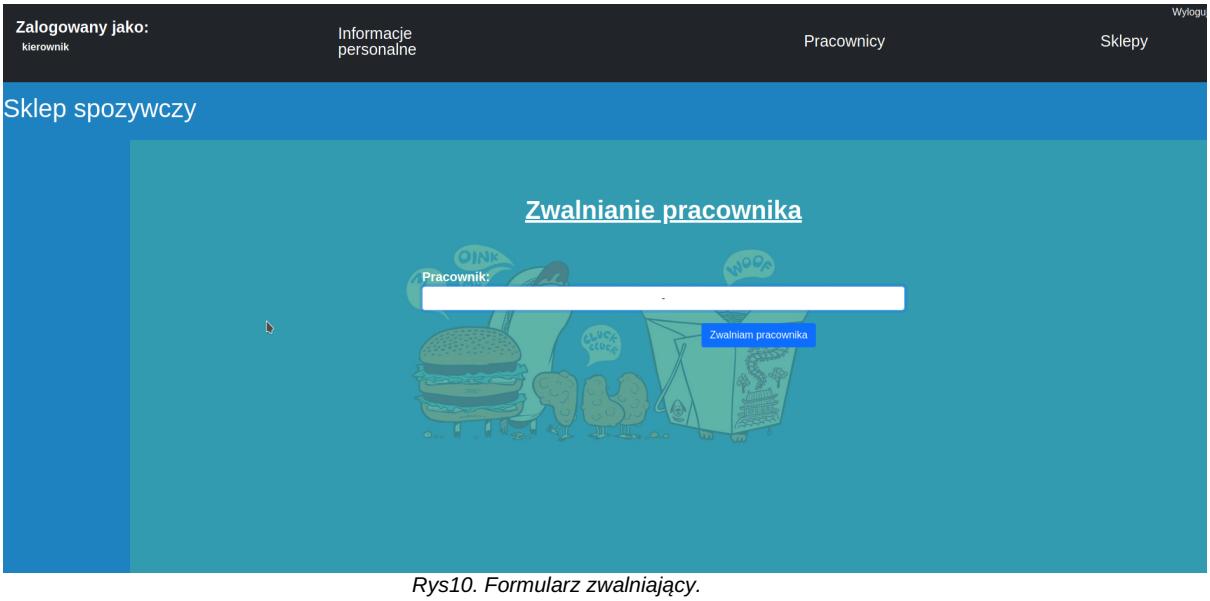
Formularz musi zostać poprawnie wypełniony przed wysłaniem. Po wybraniu opcji magazynier pojawia się możliwość wyboru sklepu sposrod tych, ktore sa “nieobstawione” przez innego pracownika. Dokonywana jest też prosta walidacja skryptem js. Po wykonaniu operacji na dole jest wyświetlany przycisk z komunikatem.

Edytuj dane - jest to kolejny formularz pozwalający zmienić dane istniejącego pracownika za wyjątkiem adresu i hasła. Pierwsze pole pozwala na wybranie spośród pracowników. Po wyborze zostaną automatycznie uzupełnione wszystkie pola, które następnie można edytować.



Rys9. Formularz edytuj dane.

Zwolnij pracownika - ukazuje się formularz pozwalający na wybranie pracownika do zwolnienia. Polega to na tym, że pracownik nie jest usuwany z bazy ale dodaje się mu datę zakończenia pracy przez co nie moze miec juz dostępu do serwisu. Dany magazyn zostaje zwolniony do puli wolnych.

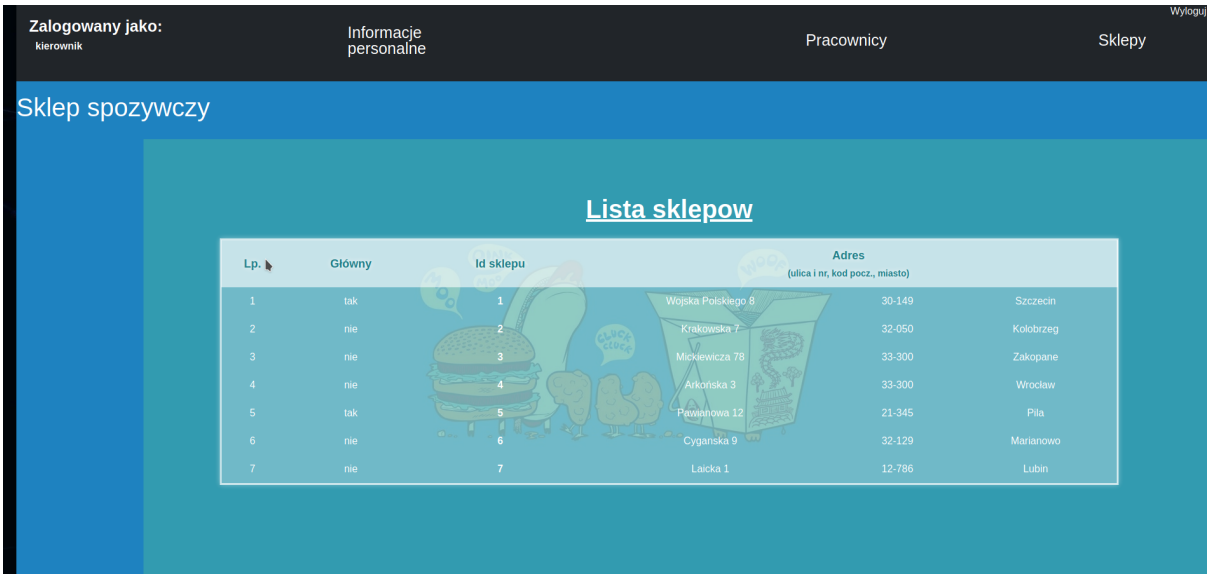


Rys10. Formularz zwalniający.

Sklepy:

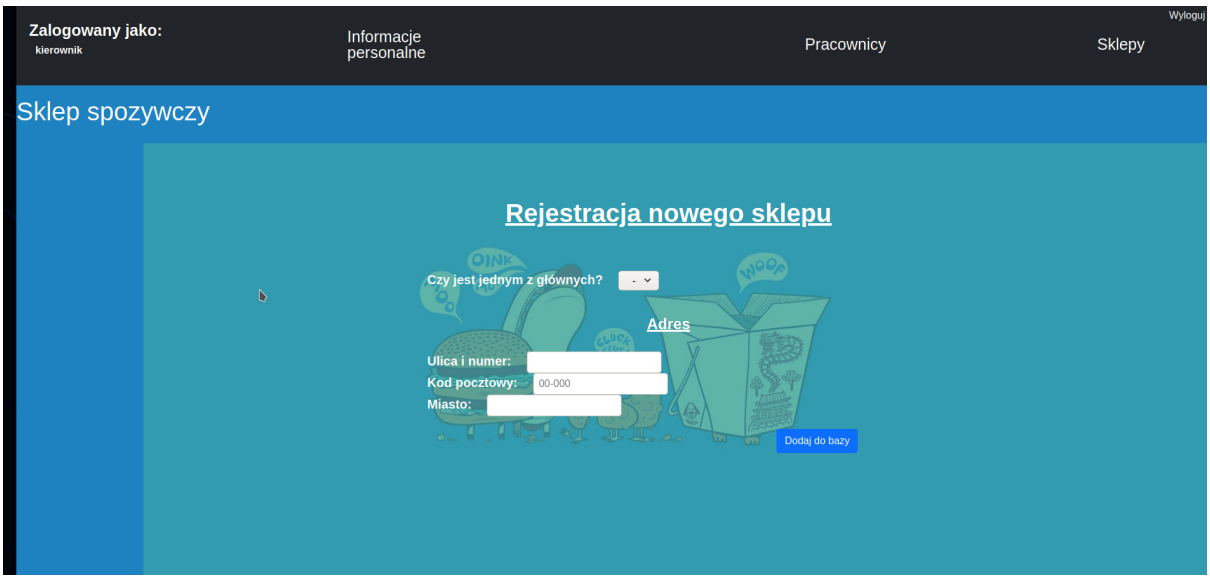
Do wyboru sa dwie opcje, lista oraz dodaj.

Lista - wyświetla listę wszystkich magazynów w formie tabeli wraz z osobami odpowiedzialnymi oraz adresem.



Rys11. Lista sklepów.

Dodaj - Zawiera formularz do dodania nowego sklepu.



Rys12. Formularz dodaj sklep.

2. *Pracownik*

Katalog produktów:

Do wyboru są dwie opcje: lista produktow oraz dodaj.

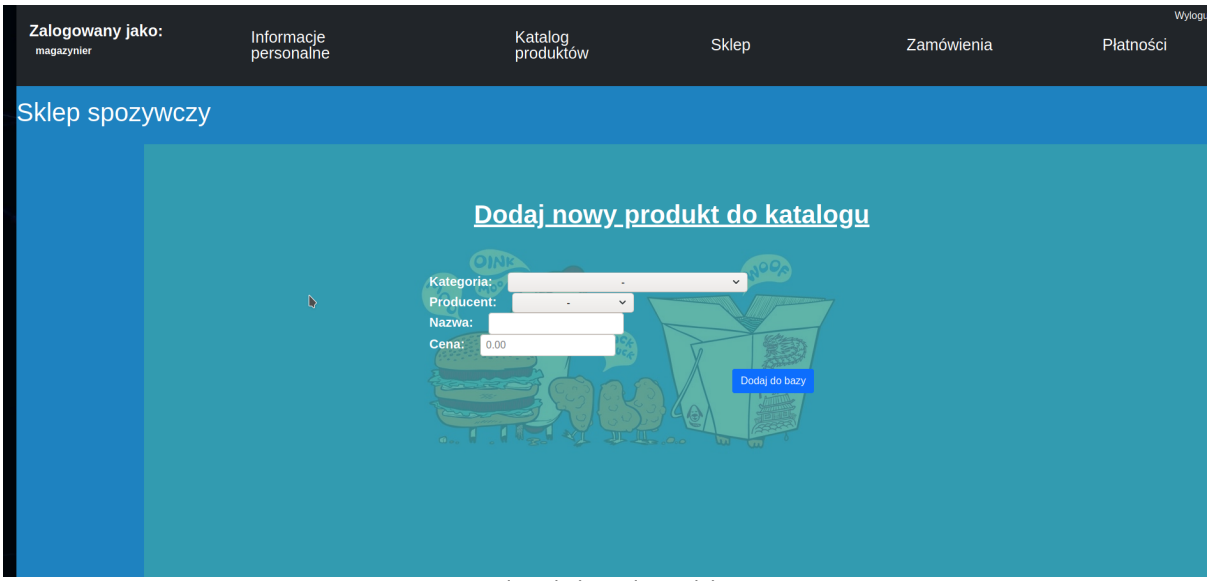
Lista produktów - Zawiera listę produktów spożywczych dla wszystkich sklepów.

Lista produktów

Lp.	Nazwa produktu	Id produktu	Kategoria	Producent	Cena
1	JABŁKO	3	Owoce dostępne na ternie kraju.	JABLKOSAD	2.39
2	SZYNKA	12	Owoce dostępne na ternie kraju.	SMACZNA SZAMKA	12.87
3	ŚLIWKI	11	Owoce dostępne na ternie kraju.	SADEX	3.22
4	POMARANCZE	10	Owoce dostępne na ternie kraju.	POMARANCZIL	3.45
5	MARCHEW	13	Owoce dostępne na ternie kraju.	WARZYWEX	12.99
6	GRUSZKA	8	Owoce dostępne na ternie kraju.	GRUSZKOPOL	4.34
7	KIWI	9	Owoce z różnych zakątków świata.	RABAR	6.91
8	BANAN	5	Owoce z różnych zakątków świata.	POMIDORO	4.55
9	POMIDOR	4	Świeże, soczyste warzywa, zielone i nie tylko.	WARZYWEX	6.77
10	WOŁOWINA - GULASZ	7	Wyrony mięsne - wieprzowina, drób, wołowina.	SMACZNA SZAMKA	15.43
11	PIERŚ Z KURCZAKA	6	Wyrony mięsne - wieprzowina, drób, wołowina.	GRUSZKOPOL	18.89

Rys13. Lista produktów.

Dodaj - pozwala pracownikowi na dodanie nowego produktu do katalogu produktów dla wszystkich sklepów. Automatycznie ilość tego produktu w poszczególnych sklepach jest ustawiana na 0.



Rys14. Formularz dodawania produktu.

Sklep:

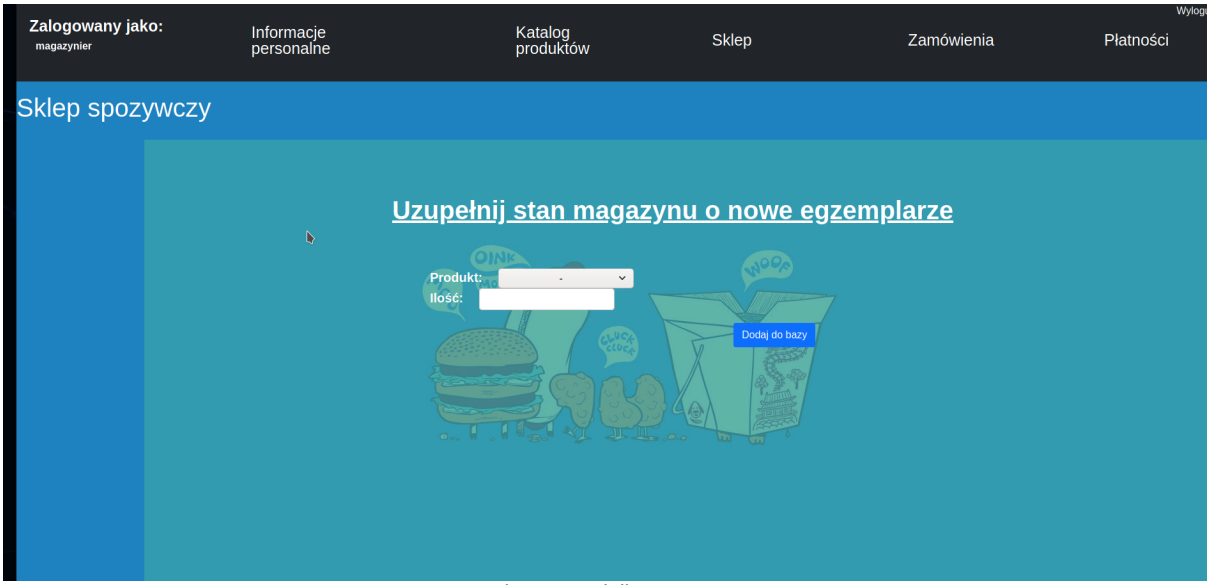
Zawiera opcje stan, uzupełnij asortyment i sprzedaż.

Stan - Zawiera listę produktów z katalogu oraz ich ilosc dostepna w sklepie. Wynik przedstawiony w formie tabeli.

Lp.	Nazwa produktu	Ilość	Cena j.	Id produktu	Kategoria	Producent
1	JABŁKO	0	2.39	3	Owoce dostępne na ternie kraju.	JABLKOSAD
2	POMIDOR	0	6.77	4	Świeże, soczyste warzywa, zielone i nie tylko.	WARZYWEX
3	BANAN	0	4.55	5	Owoce z różnych zakątków świata.	POMIDORO
4	PIERŚ Z KURCZAKA	0	18.89	6	Wyrony mięsne - wieprzowina, drób, wołowina.	GRUSZKOPOL
5	WOŁOWINA - GULASZ	0	15.43	7	Wyrony mięsne - wieprzowina, drób, wołowina.	SMACZNA SZAMKA
6	GRUSZKA	0	4.34	8	Owoce dostępne na ternie kraju.	GRUSZKOPOL
7	KIWI	0	6.91	9	Owoce z różnych zakątków świata.	RABAR
8	POMARANCZE	6	3.45	10	Owoce dostępne na ternie kraju.	POMARANCZIL
9	ŚLIWKI	1	3.22	11	Owoce dostępne na ternie kraju.	SADEX
10	SZYNKA	54	12.87	12	Owoce dostępne na ternie kraju.	SMACZNA SZAMKA
11	MARCHEW	22	12.99	13	Owoce dostępne na ternie kraju.	WARZYWEX

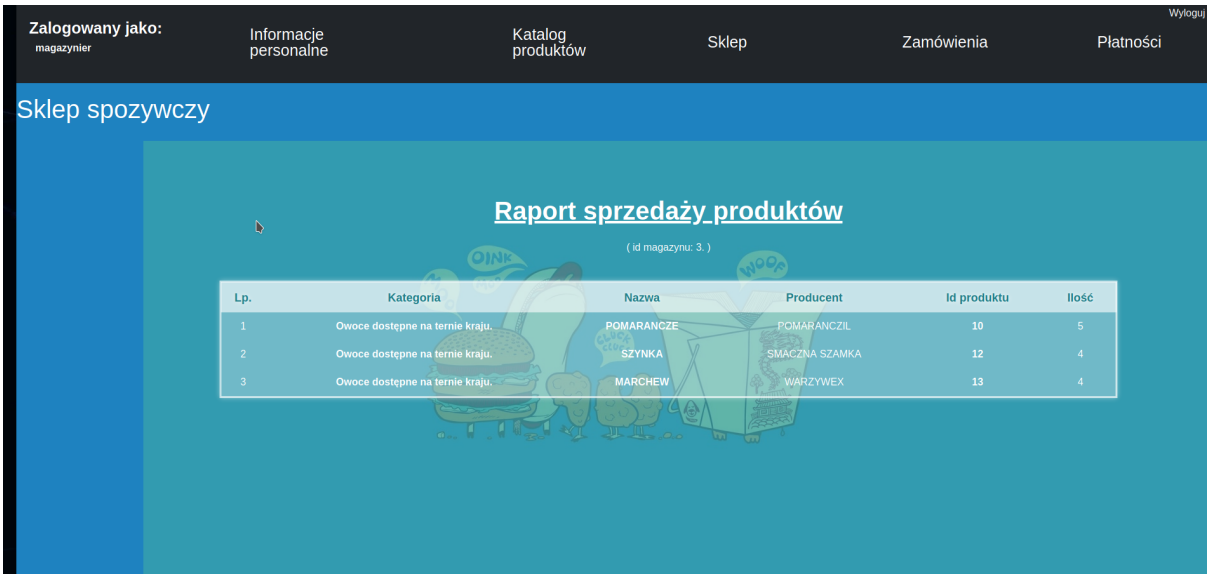
Rys15. Lista stanu sklepu.

Uzupełnij asortyment - Formularz pozwalający na uzupełnienie asortymentu i dodanie konkretnej ilości artykuły do sklepu.



Rys16. Formularz uzupełnij asortyment.

Sprzedaż - w formie tabeli widoczny jest raport sprzedaży konkretnego sklepu.

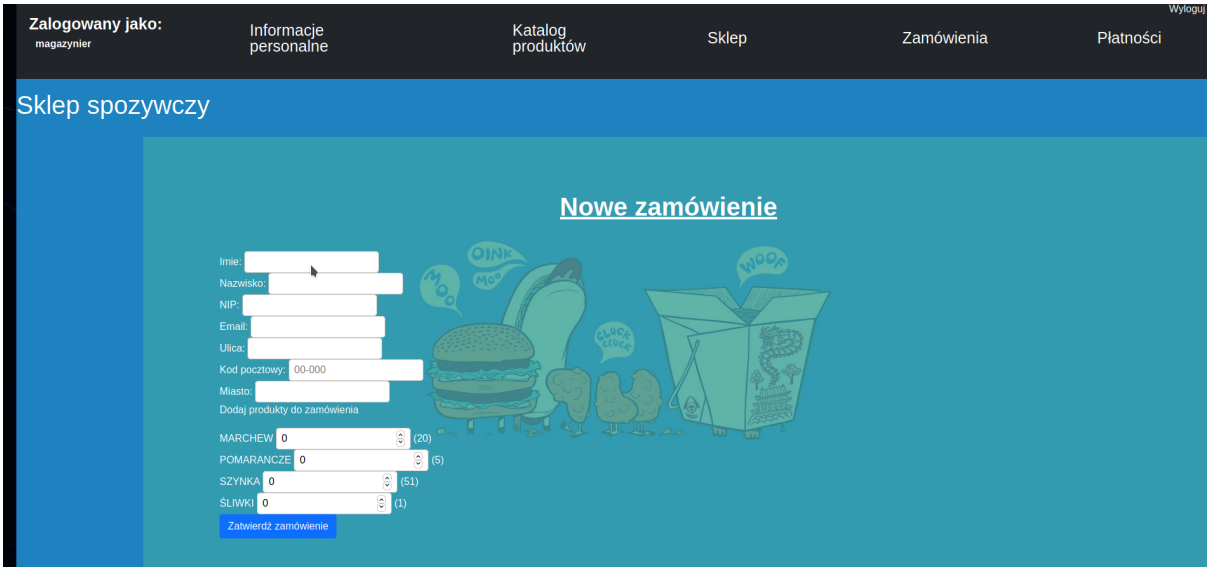


Rys17. Lista sprzedanych artykułów przez sklep.

Zamówienia:

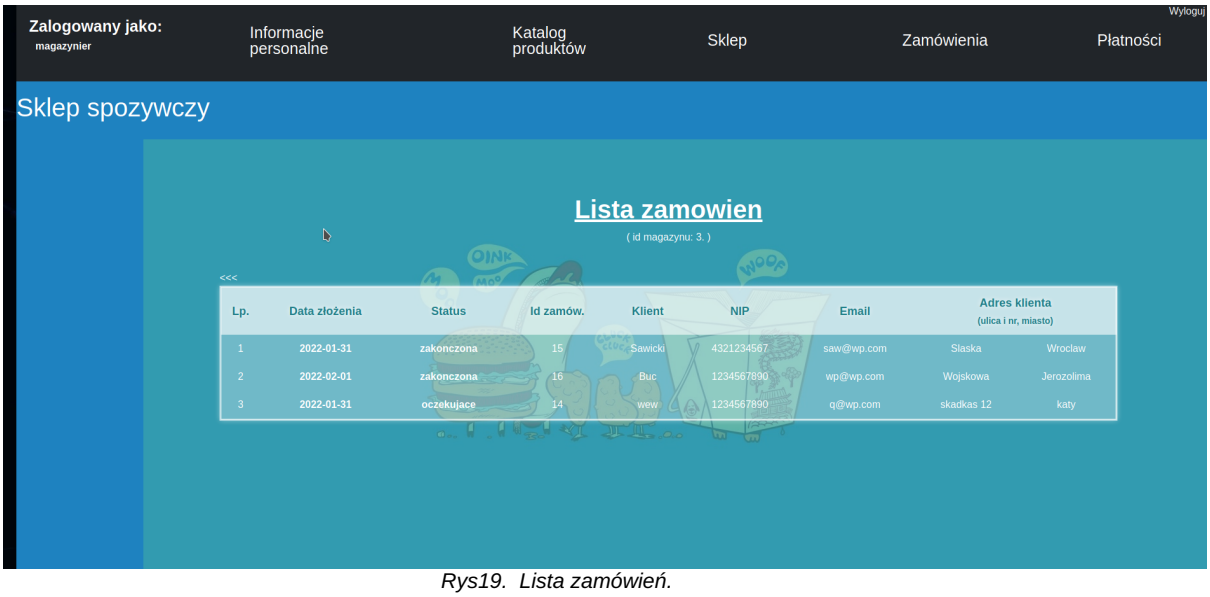
Zawiera opcje Nowe zamówienie, Lista.

Nowe zamówienie - dostępny jest formularz umożliwiający dodanie nowego zamówienia wraz z przypisanym mu klientem. Pozwala na wybranie produktów z tych dostępnych w konkretnym sklepie.



Rys18. Formularz nowe zamówienie.

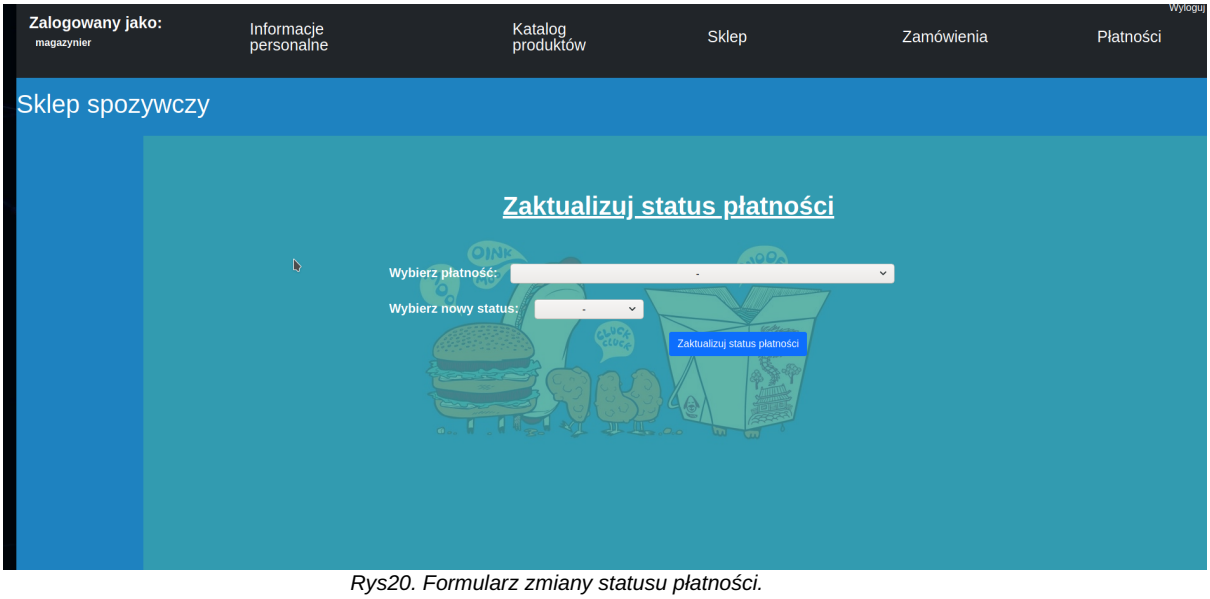
Lista - wyświetla wszystkie zamówienia w formie tabeli.



Rys19. Lista zamówień.

Płatności:

Zawiera opcje aktualizuj płatność. Po wyborze zamówienia z listy możemy zaktualizować mu status płatności.



Rys20. Formularz zmiany statusu płatności.

Dla obu wersji użytkowników dostępna jest opcja wyloguj, która przekierowuje na stronę główna do logowania.

V. Dokumentacja

16. Wprowadzanie danych:

W większości przypadków jest ręcznie za pomocą formularza. Niektóre dane są ustalane z góry przez triggery tak jak wartość ilości danego produktu w sklepie.

17. Dokumentacja użytkownika: krótka instrukcja obsługi aplikacji:

Informacje pozwalające poruszać się po aplikacji są zawarte w paragrafie IV.

Dodatkowo dane do logowanie:

- kierownik
  - haslo: nowak
  - login: nowak
- pracownik
  - haslo: maly
  - login: maly

18. Opracowanie dokumentacji technicznej:

Krótkie komentarze opisowe znajdują się w plikach źródłowych.

19. Wykaz literatury:

https://www.lucidchart.com

https://stackoverflow.com
https://newton.fis.agh.edu.pl/~antek/index.php
https://www.php.net/docs.php
https://www.postgresql.org/docs/