

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

PROJEKT Z BAZ DANYCH

Sklep Internetowy

Termin zajęć: Środa, 13:15–15:00

AUTORZY:

Mateusz Śmiechura	235586	235586@student.pwr.edu.p
Sebastian Faber	235783	235783@student.pwr.edu.p
Tomasz Ławski	235122	235122@student.pwr.edu.p

PROWADZĄCY ZAJĘCIA:

dr inż. Roman Ptak, W4/K9

Wrocław, 2019 r.

Spis treści:

1. Wstęp	4
1.1. Cel projektu	4
1.2. Zakres projektu	4
2. Analiza wymagań	4
2.1. Opis działania i schemat logiczny systemu	4
2.2. Wymagania funkcjonalne	4
2.3. Wymagania нефunkcjonalne	4
2.3.1. Wykorzystywane technologie i narzędzia	4
2.3.2. Wymagania dotyczące rozmiaru bazy danych	4
2.3.3. Wymagania dotyczące bezpieczeństwa systemu	4
2.4. Przyjęte założenia projektowe	4
3. Projekt systemu	4
3.1. Projekt bazy danych	4
3.1.1. Analiza rzeczywistości i uproszczony model konceptualny	4
3.1.2. Model logiczny i normalizacja	4
3.1.3. Model fizyczny i ograniczenia integralności danych	4
3.1.4. Inne elementy schematu – mechanizmy przetwarzania danych	4
3.1.5. Projekt mechanizmów bezpieczeństwa na poziomie bazy danych	4
3.2. Projekt aplikacji użytkownika	4
3.2.1. Architektura aplikacji i diagramy projektowe	4
3.2.2. Interfejs graficzny i struktura menu	4
3.2.3. Projekt wybranych funkcji systemu	4
3.2.4. Metoda podłączania do bazy danych – integracja z bazą danych	4
3.2.5. Projekt zabezpieczeń na poziomie aplikacji	4
4. Implementacja systemu baz danych	4

<u>4.1. Tworzenie tabel i definiowanie ograniczeń</u>	4
<u>4.2. Implementacja mechanizmów przetwarzania danych</u>	5
<u>4.3. Implementacja uprawnień i innych zabezpieczeń</u>	5
<u>4.4. Testowanie bazy danych na przykładowych danych</u>	5
<u>5. Implementacja i testy aplikacji</u>	5
<u>5.1. Instalacja i konfigurowanie systemu</u>	5
<u>5.2. Instrukcja użytkowania aplikacji</u>	5
<u>5.3. Testowanie opracowanych funkcji systemu</u>	5
<u>5.4. Omówienie wybranych rozwiązań programistycznych</u>	5
<u>5.4.1. Implementacja interfejsu dostępu do bazy danych</u>	5
<u>5.4.2. Implementacja wybranych funkcjonalności systemu</u>	5
<u>5.4.3. Implementacja mechanizmów bezpieczeństwa</u>	5
<u>6. Podsumowanie i wnioski</u>	5
<u>Literatura</u>	5
<u>Spis rysunków</u>	5
<u>Spis tabel</u>	5

1. Wstęp

1.1. Cel projektu

Stworzenie sklepu internetowego sprzedającego części elektroniczne oraz panelu administratora, umożliwiającego zarządzanie sklepem.

1.2. Zakres projektu

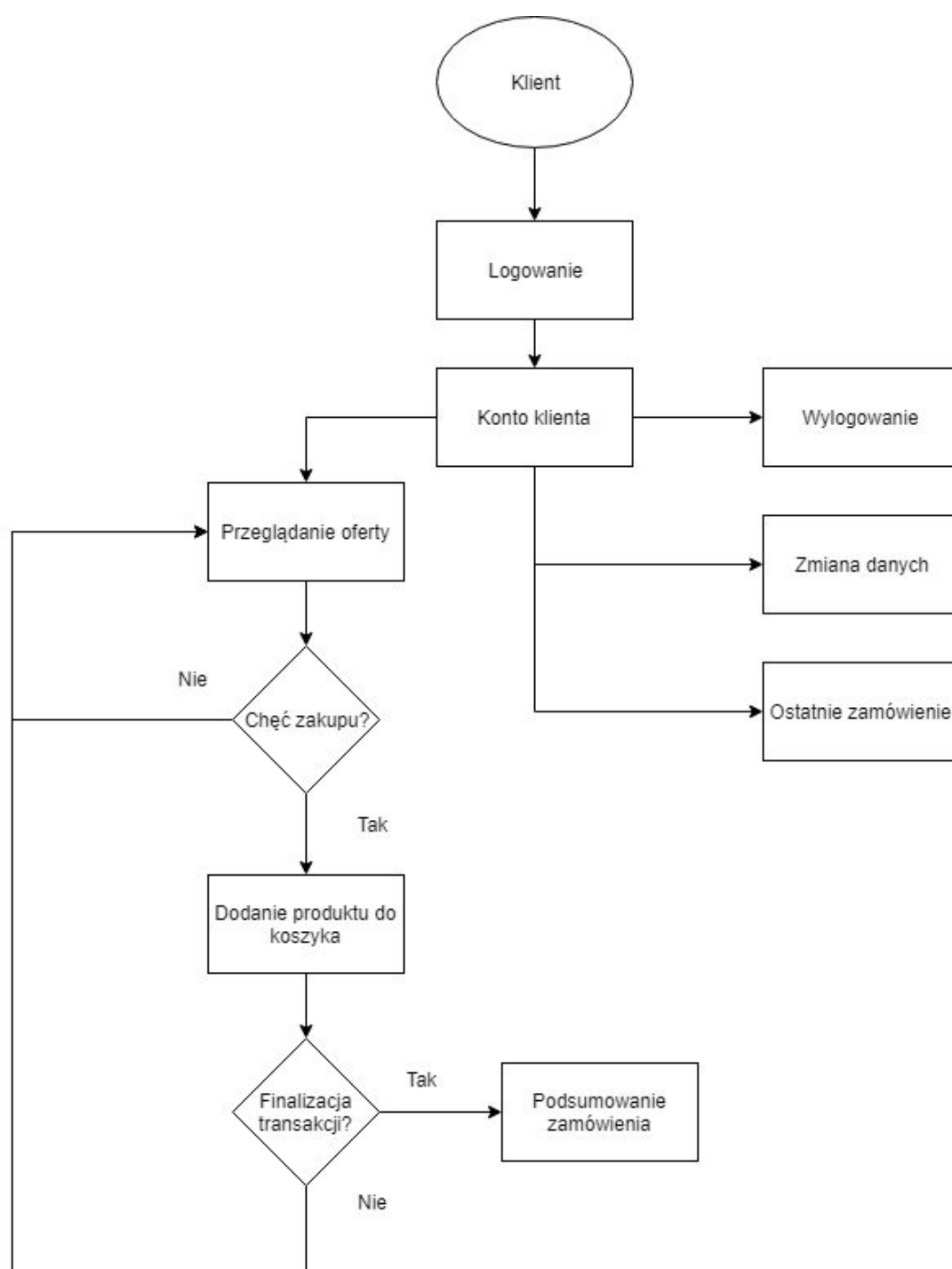
- Stworzenie bazy danych sklepu
- Stworzenie aplikacji dostępowej klienta
- Stworzenie aplikacji dostępowej administratora
- Stworzenie dokumentacji projektu

2. Analiza wymagań

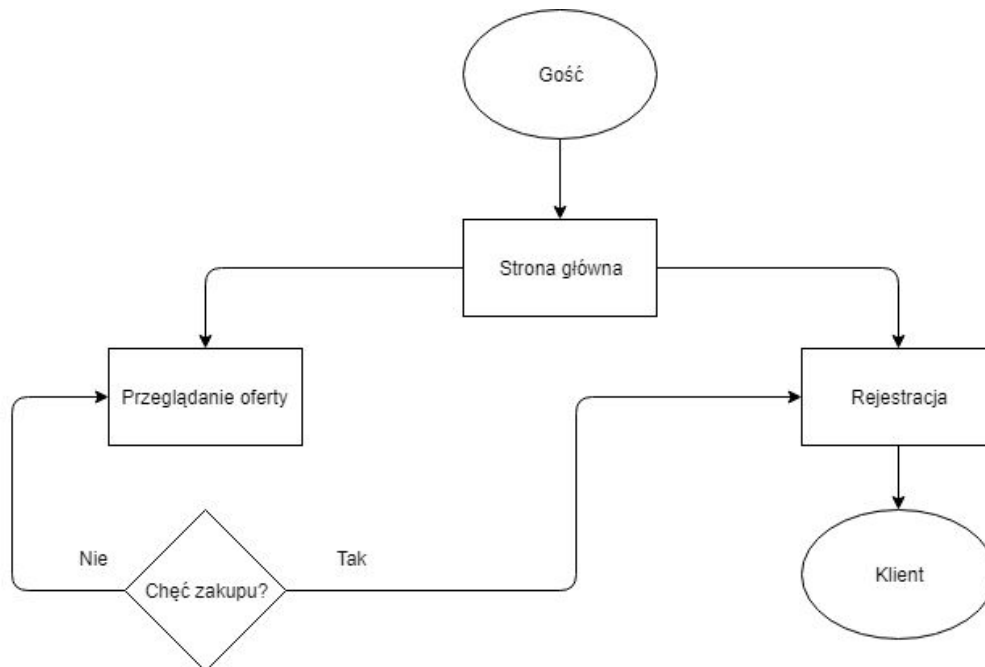
2.1. Opis działania i schemat logiczny systemu

Aplikacja będzie miała postać sklepu internetowego firmy MOHITwrO, która zajmuje się sprzedażą części elektronicznych. Firma posiada właścicieli oraz klientów. Każdy użytkownik ma możliwość przeglądania asortymentu sklepu, natomiast po zarejestrowaniu zyskuje możliwość zakupienia produktów z dostępnej oferty, przeglądania swoich ostatnich zamówień oraz zmiany swoich danych osobowych. Administrator systemu będzie miał możliwość zarządzania tabelą produktów, przeglądania tabeli klientów oraz zarządzania tabelą zamówień. Oferta sklepu jest uprzednio dodana przez administratora systemu.

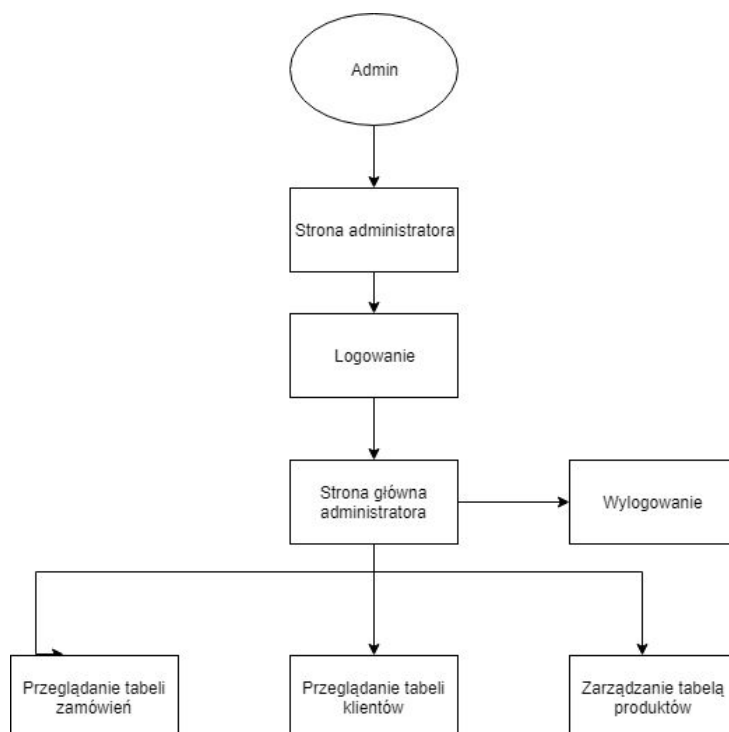
Schemat przedstawiający możliwe interakcje klienta sklepu z serwisem:



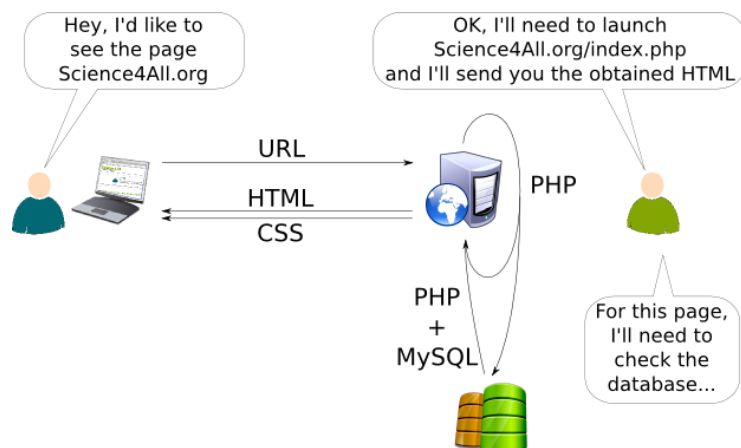
Schemat przedstawiający możliwe interakcje gościa sklepu z serwisem:



Schemat przedstawiający możliwe interakcje administratora systemu z serwisem:



Schemat przedstawiający komunikację aplikacji z użytkownikiem:



2.2. Wymagania funkcjonalne

A. Administrator - właściciel.

Uprawnienia:

- Dodawanie towarów na sprzedaż
- Edytowanie dostępnych towarów
- Modyfikowanie/przeglądanie tabeli towarów
- Przeglądanie tabeli klientów
- Przeglądanie tabeli zamówień
- Zmiana statusu zamówień niezrealizowanych
- Dodawanie/usuwanie/modyfikowanie/przeglądanie kategorii przedmiotów
- Logowanie do systemu

B. Gość - użytkownik niezarejestrowany.

Posiada:

- możliwość przeglądania produktów
- możliwość wyszukiwania produktów (filtrowanie)
- możliwość rejestracji

C. Klient - użytkownik zarejestrowany i zalogowany, posiada uprawnienia i możliwości gościa oraz dodatkowe, wyszczególnione niżej, cechy oraz możliwości. Dokonuje rejestracji samodzielnie.

Uprawnienia:

- Przeglądanie tabeli towarów
- Zamówienie wybranych towarów
- Przeglądanie oraz sprawdzanie statusu swoich ostatnich zamówień
- Edycję swoich danych osobowych
- wyszukiwanie produktu po nazwie lub kategorii
- dodawanie oraz usuwanie zamówień do/z koszyka
- możliwość logowania do serwisu

2.3. Wymagania нефunkcjonalne

A. Administrator - właściciel.

Posiada:

- Imię
- Nazwisko
- Login
- Hasło - posiadające od 8 do 24 znaków, bez znaków specjalnych

B. Przedmiot - obiekt przeznaczony na sprzedaż zarejestrowanym użytkownikom.

Posiada:

- Numer ID - unikalny dla każdego przedmiotu. Umożliwiający dostęp do konkretnego produktu
- Kategorię - określającą przynależność do grupy produktów
- Nazwę
- Cechę - umożliwiającą filtrowanie wyników
- Cenę - określającą wartość produktu, wyrażoną z dokładnością do 0,01 zł
- Grafikę - przedstawiającą dany przedmiot
- Stan magazynowy - określający dostępną ilość produktu na magazynie
- Opis - specyfikacja techniczna produktu

C. Klient - użytkownik zarejestrowany.

Posiada:

- Numer ID - unikalny dla każdego klienta. Umożliwiający stworzenie relacyjności z zamówieniami
- Imię
- Nazwisko
- Login - unikalny i niepowtarzalny w bazie
- Hasło - posiadające od 8 do 24 znaków, bez znaków specjalnych.
- E-mail - w formie *przykładowy_mail@domena.pl*, co będzie sprawdzane
- Kod pocztowy - dane do wysyłki
- Miasto pocztowe - dane do wysyłki
- Adres - dane do wysyłki
- Telefon - posiadający 9 cyfr, co będzie sprawdzane

D. Zamówienie - zamówienie złożone przez zarejestrowanego klienta, do realizacji przez administratora.

Posiada:

- Numer ID - unikalny dla każdego zamówienia
- Numer zamówienia - unikalny dla każdej grupy zamówień. Umożliwiający połączenie dowolnej ilości zamówień w grupę zamówień
- Numer ID zamówionego przedmiotu - połączenie relacyjne z tabelą przedmiotów
- Ilość zamówionego przedmiotu
- Numer ID zamawiającego klienta - połączenie relacyjne z tabelą klientów
- Data i godzina zamówienia - Data i godzina, w której klient potwierdził zamówienie
- Wybrany przewoźnik - dostawca zamówienia {FedEx, UPS, DHL, Poczta Polska, DPD, Pocztex, odbiór osobisty}
- Status zamówienia {złożone, w realizacji, przygotowane do wysyłki, wysłane}

2.3.1. Wykorzystywane technologie i narzędzia

- Platforma sprzętowa: urządzenie z dostępem do internetu
- Implementacja bazy danych: MySQL
- Praca z bazą danych: PHP
- Oprawa merytoryczna strony: HTML5
- Oprawa graficzna strony: CSS

- Narzędzie programistyczne: Notepad++
- Serwer lokalny Apache przy użyciu programu XAMPP
- Modelowanie baz danych: ERD
- Rysowanie schematów blokowych: Draw.io

2.3.2. Wymagania dotyczące rozmiaru bazy danych

Planowanych jest 3 administratorów. Początkowa liczba produktów będzie wynosić około 100, lecz zakładamy, iż baza produktów będzie, w niedalekiej przyszłości, rozbudowywana do liczby około 1000 elementów. Początkowa liczba klientów będzie wynosić 0, lecz zakładamy, że w ciągu pierwszego miesiąca zarejestruje się około 5000 osób, a ich liczba będzie rosła z miesiąca na miesiąc do szacowanej liczby 50 tysięcy klientów. Rozmiar bazy danych zamówień początkowo będzie zerowy, jednak przy dużym nakładzie klientów szacujemy, iż baza po pierwszym miesiącu osiągnie przynajmniej 50 tysięcy zamówień, a w przyszłości osiągnie szacowaną liczbę 2 milionów.

2.3.3. Wymagania dotyczące bezpieczeństwa systemu

- Dostęp do panelu administratora będzie posiadał tylko zalogowany administrator, któremu twórca sklepu stworzył konto administratorskie i podał mu dane logowania. Po zalogowaniu administrator może zmienić swoje hasło. Aby otworzyć panel administratora, administrator będzie musiał wpisać w adresie strony, po głównej domenie, specjalny 24-znakowy kod.
- Dostęp do panelu klienta posiadać będzie tylko zalogowany klient. Ma on dostęp do wszystkich swoich danych wpisanych podczas rejestracji.
- Tabele administratorów niedostępna dla nikogo.
- Tabele przedmiotów dostępna publicznie.
- Tabele zamówień dostępna tylko w panelu administratora.
- Imię, nazwisko oraz e-mail klientów dostępne tylko dla administratora.
- Hasło klienta zostanie przepuszczone przez funkcję hash-ującą, co uniemożliwi zarządcy systemu wgląd do haseł użytkowników.
- Zabezpieczenia filtrujące dane z wpisywanych formularzy, aby zapobiec włamaniom do bazy danych (SQL Injection).

2.4. Dodatkowe założenia projektowe

Uogólnione założenia:

- Przejrzystość strony
- Prostota w użytkowaniu strony
- Odporność na włamania do systemu
- Odporność na błędy użytkownika (takie jak wprowadzenie adresu email w niepoprawnej formie)

3. Projekt systemu

Projekt i struktury bazy danych, mechanizmów zapewniania poprawności przechowywanych informacji, oraz kontroli dostępu do danych.

3.1. Projekt bazy danych

3.1.1. Analiza rzeczywistości i uproszczony model konceptualny

Program MySQL Workbench

3.1.2. Model logiczny i normalizacja

Normalizacja 3NF - konwersja nieznormalizowanych danych w tabele; podział tabel bez straty powiązań; każda tabela posiada klucz główny PRIMARY.

3.1.3. Model fizyczny i ograniczenia integralności danych

Integralność semantyczna, Integralność encji, Integralność referencyjna

3.1.4. Inne elementy schematu – mechanizmy przetwarzania danych

- Indeksy

admin - id_admin PRIMARY, login

kategorie - id_kat PRIMARY, nazwa

producenci - id_producent PRIMARY, nazwa

produkty - id_produkt PRIMARY, kategoria FOREIGN, producent FOREIGN, cecha, cena,

dostepnosc

det_zam - id_zam_pro PRIMARY, nr_zamow FOREIGN, nr_prod FOREIGN

adresy - id_adres PRIMARY, kod FOREIGN

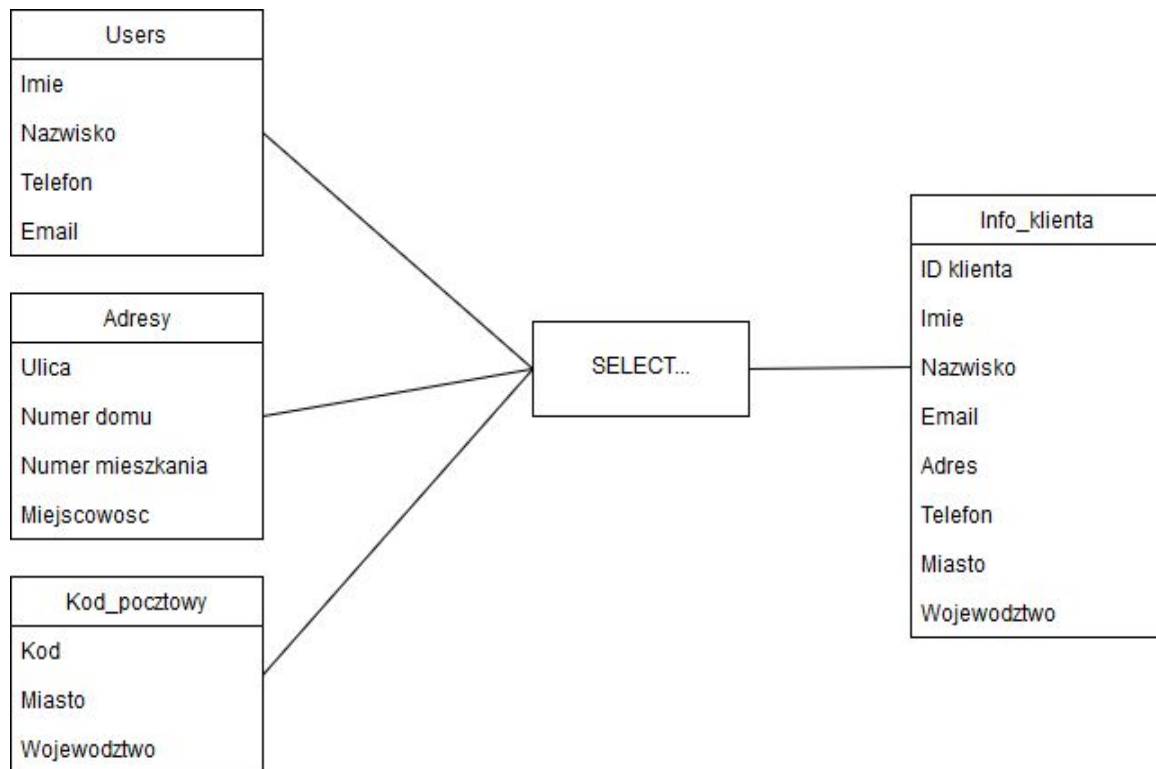
kod_pocztowy - id_kod PRIMARY

zamowienia - id_zamowienia PRIMARY, klient FOREIGN, data_zam, status

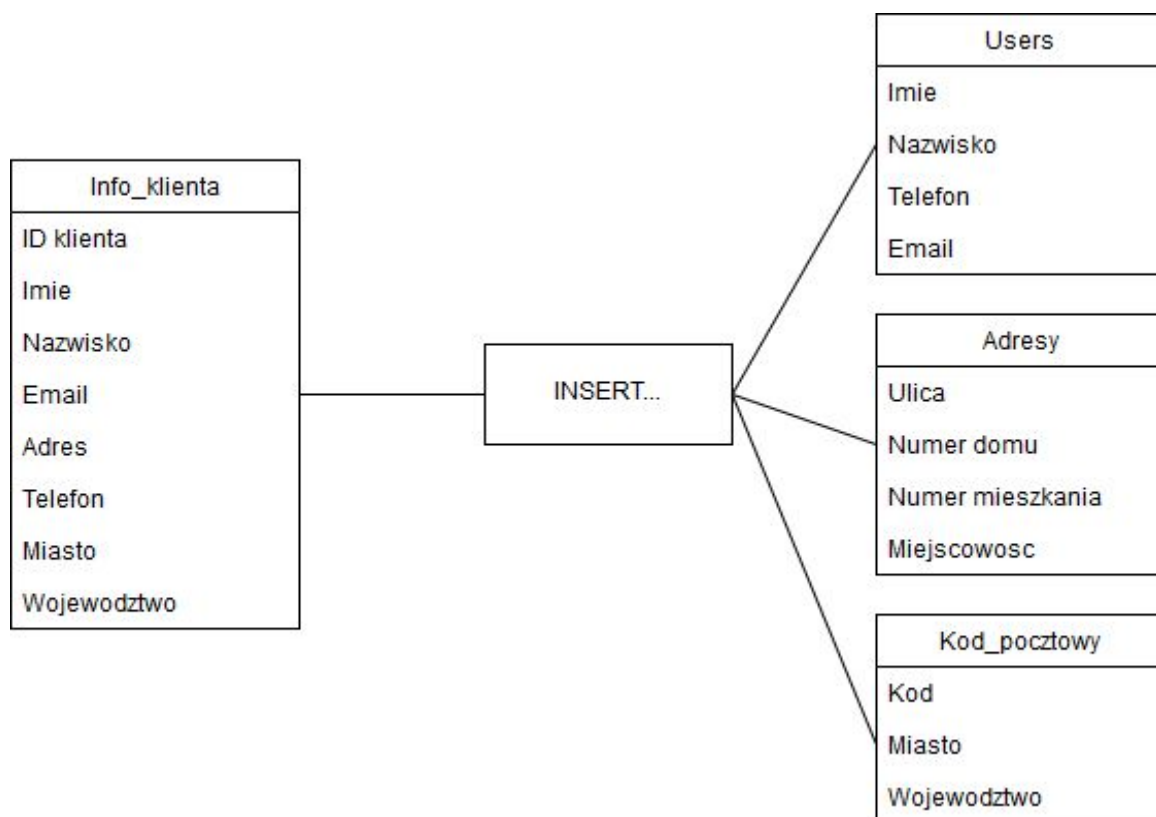
users - id_user PRIMARY, adres FOREIGN, imie + nazwisko, login

- Widoki

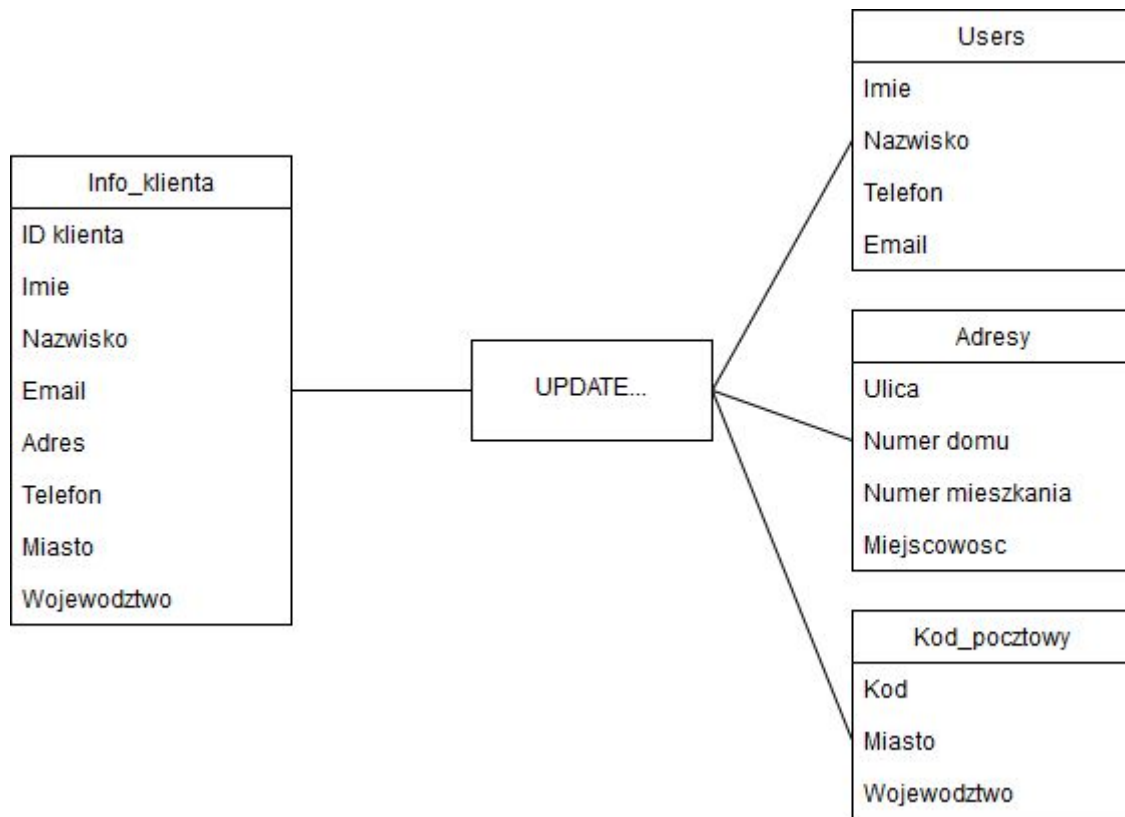
- informacje o użytkowniku SELECT



- dodaj uzytkownika dla goscia



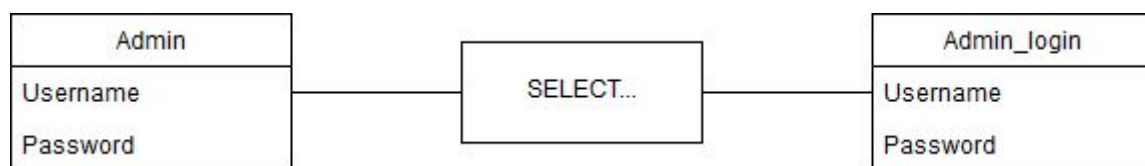
- edytuj dane dla uzytkownika



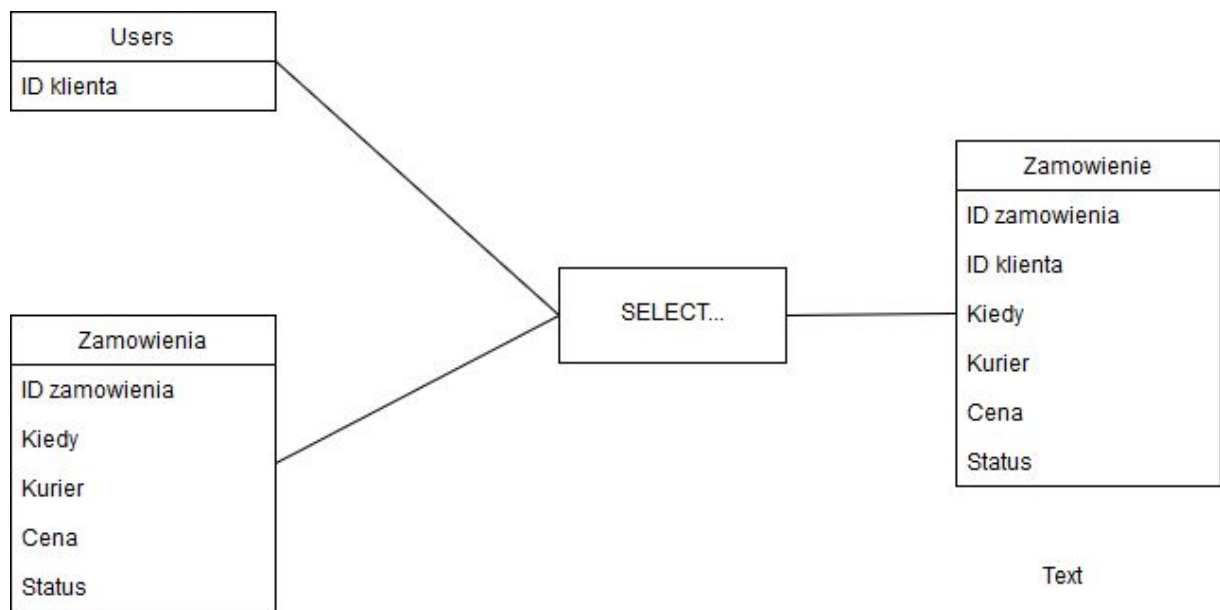
- logowanie uzytkownika SELECT



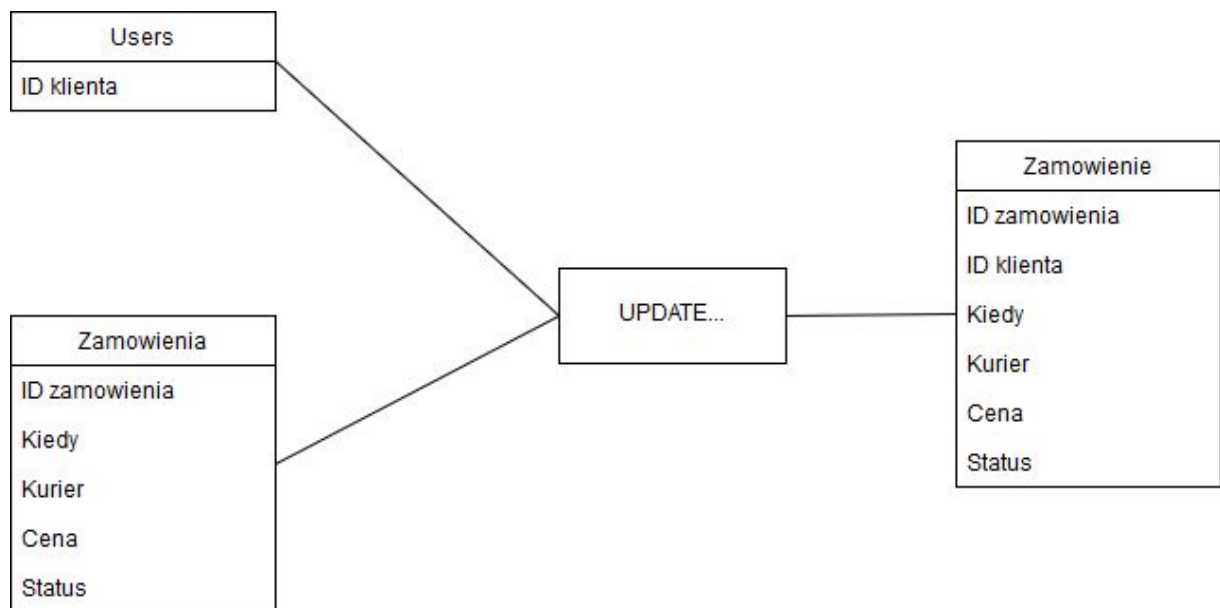
- administratora SELECT



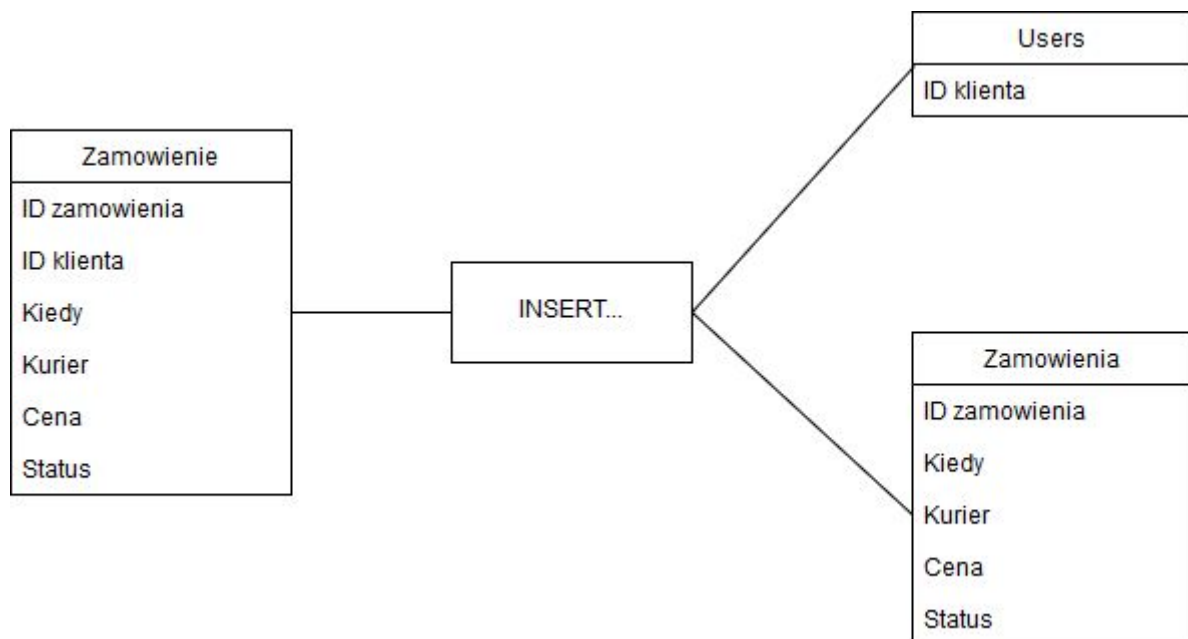
- zamowienie - informacje SELECT



- zamowienie - edytuj status



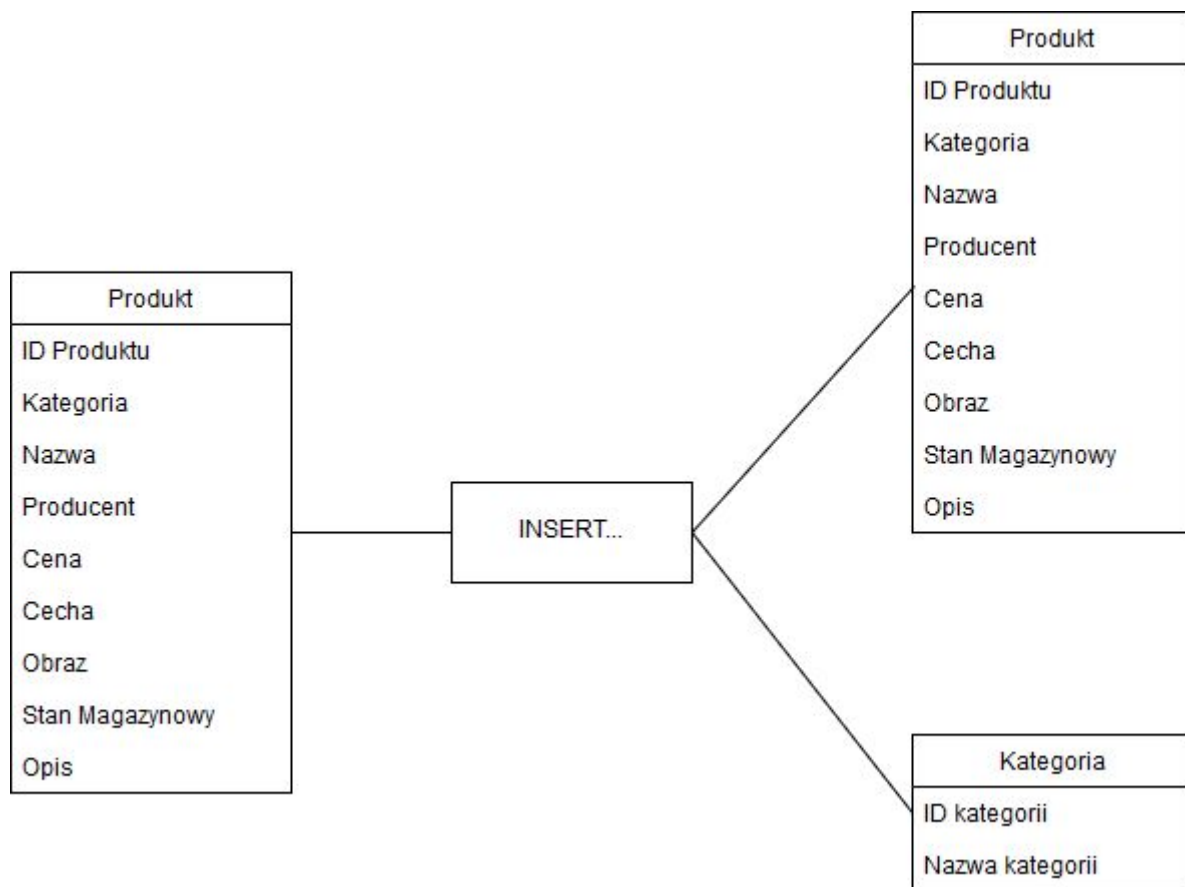
- dodaj zamowienie



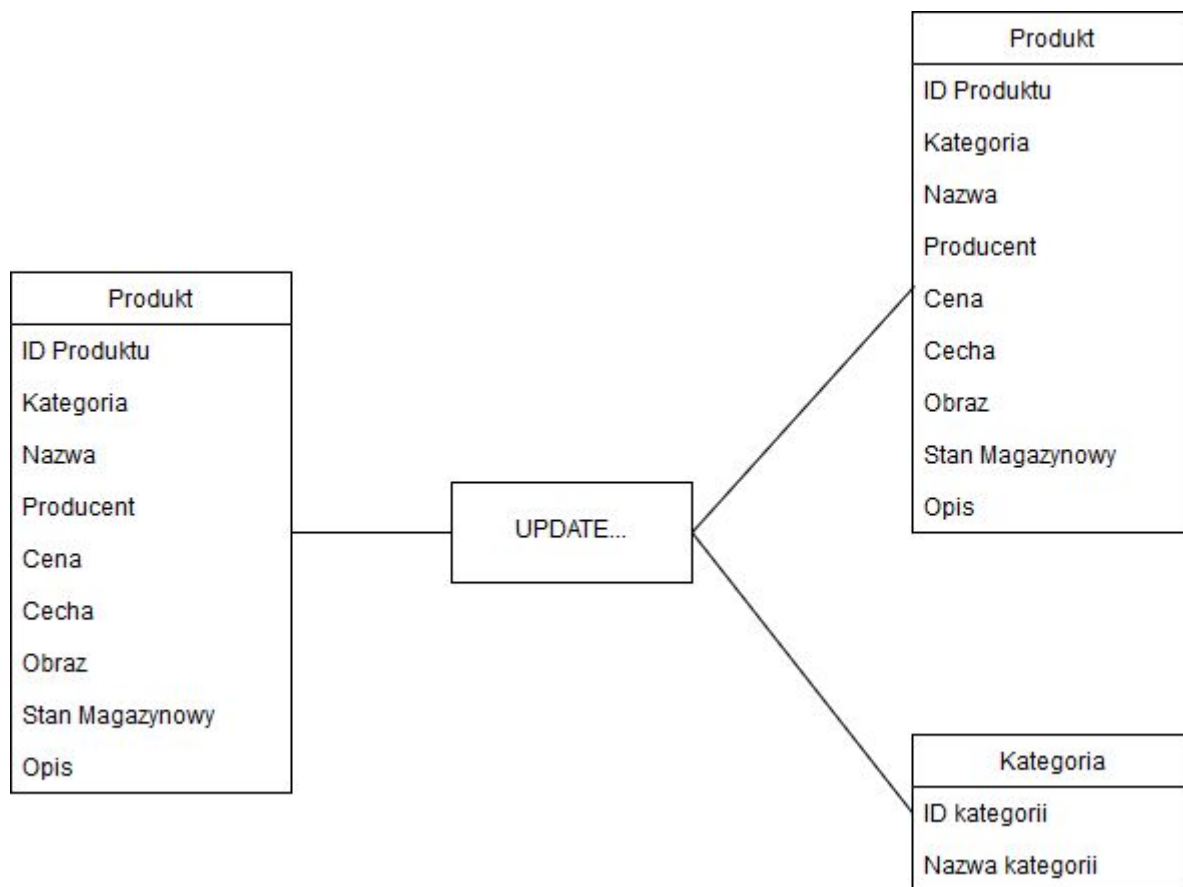
- kategorie - SELECT wszystkie



- dodaj produkt INSERT

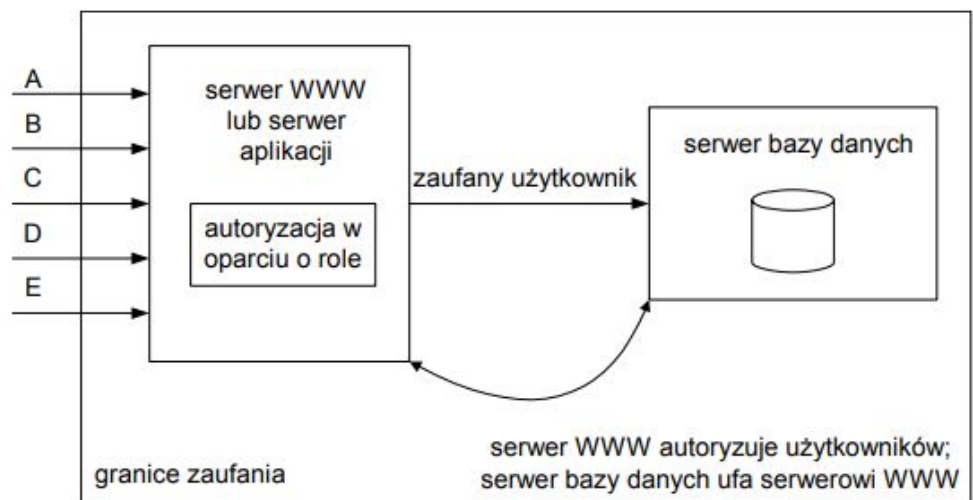


- edytuj produkt UPDATE



3.1.5. Projekt mechanizmów bezpieczeństwa na poziomie bazy danych

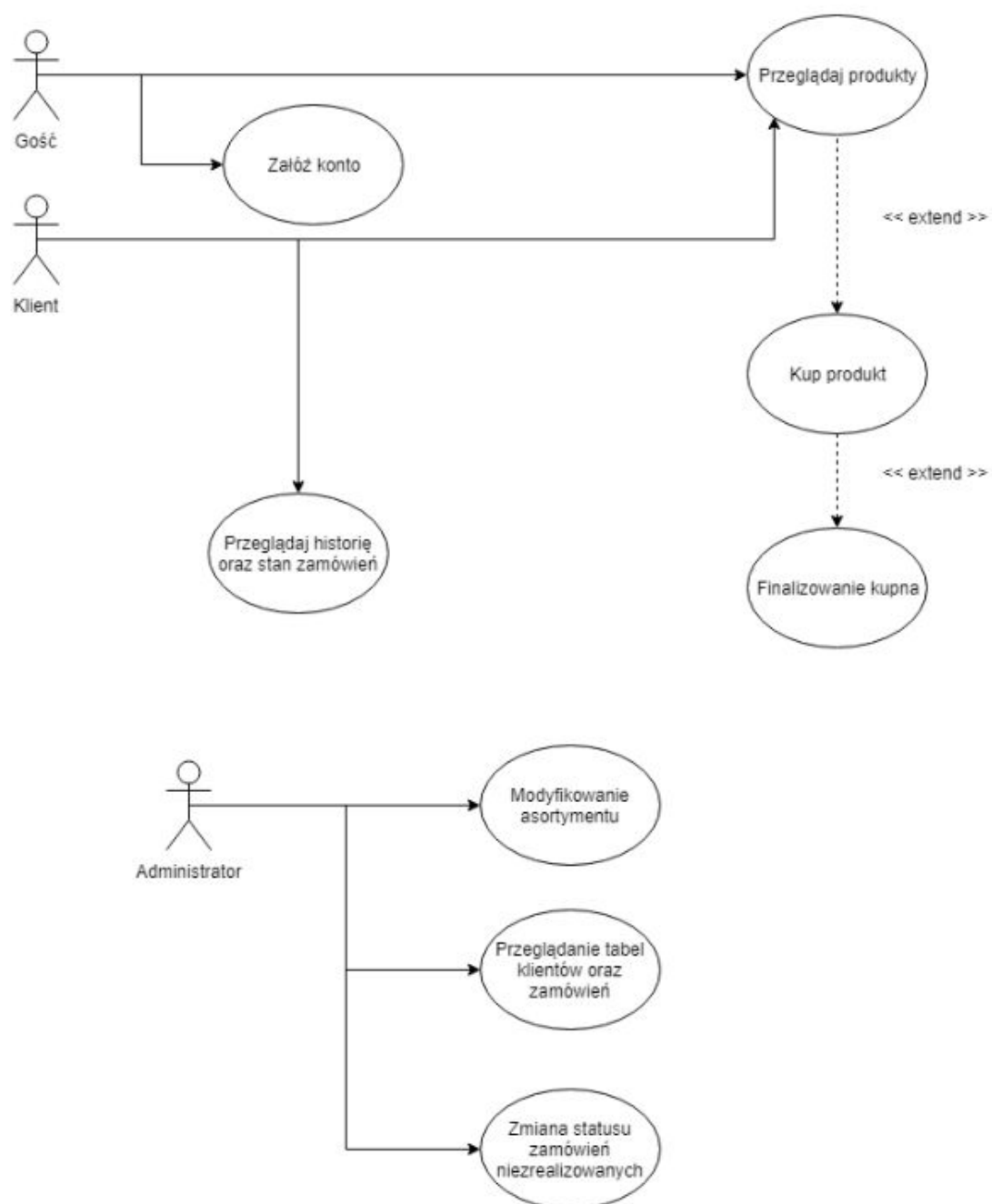
- Umożliwienie korzystania z serwera tylko osobom do tego uprawnionym (login i hasło)
- podział poziomów dostępu do bazy danych dla różnych użytkowników (Klient - dane osobowe (tylko własne)/dane produktów, Gość - dane produktów, Admin - dane produktów, spis zamówień),
- Uwierzytelnienie na poziomie serwera (po stronie serwera sprawdzane są poprawności danych logowania).
- Uwierzytelnienie na poziomie bazy danych (użytkownik bazy danych ma dostęp do konkretnej tabeli - w bazie danych utworzony jest spis uprawnień dostępu poszczególnych użytkowników (dla uzyskania uprawnień gościa nie potrzebne jest uwierzytelnienie)).



Szyfrowanie haseł użytkowników (w bazie danych przechowywane są jako zaszyfrowany ciąg znaków).

3.2. Projekt aplikacji użytkownika

3.2.1. Architektura aplikacji i diagramy projektowe

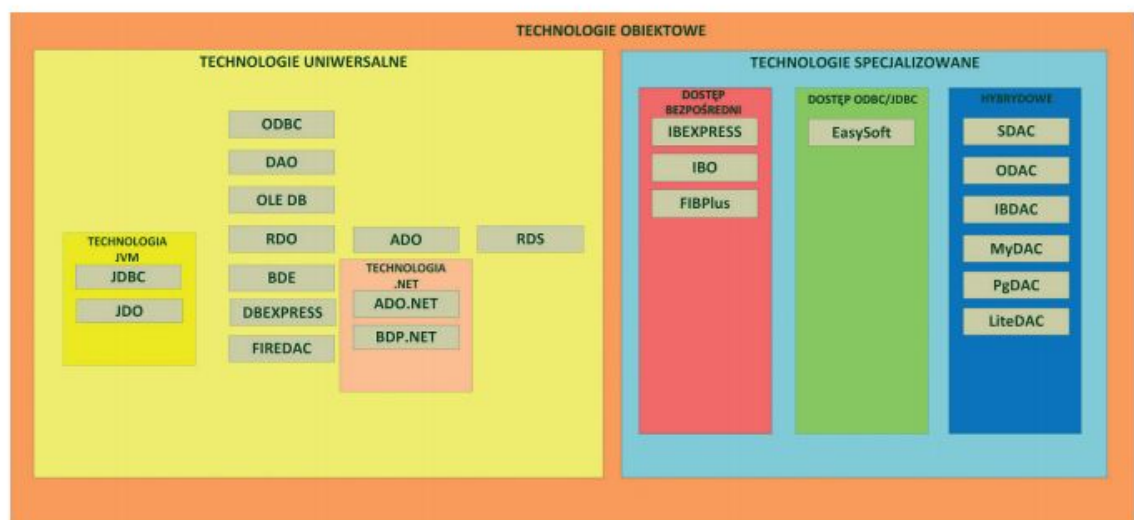


3.2.2. Interfejs graficzny i struktura menu

LOGO	WYSZUKIWARKA			PANEL UŻYTKOWNIKA / GOŚCIA
KATEGORIE	PRODUKT	PRODUKT	PRODUKT	PRODUKT
	PRODUKT	PRODUKT	PRODUKT	PRODUKT
	PRODUKT	PRODUKT	PRODUKT	PRODUKT
	PRODUKT	PRODUKT	PRODUKT	PRODUKT
	PRODUKT	PRODUKT	PRODUKT	PRODUKT
	PRODUKT	PRODUKT	PRODUKT	PRODUKT

3.2.3. Metoda podłączania do bazy danych – integracja z bazą danych

Połączenie realizowane jest przy pomocy modułu mysql.



3.2.4. Projekt zabezpieczeń na poziomie aplikacji

- Umożliwienie podjęcia próby zakupu tylko osobom zarejestrowanym i zalogowanym w celu uniemożliwienia wysyłania wielu zapytań (żądających kupna danego produktu) do serwera,
- Dostęp do bazy danych tylko i wyłącznie z poziomu aplikacji (w celu ochrony samej bazy),

- Szyfrowanie(funkcją Hash) hasła użytkownika przed późniejszym porównaniem go z oryginałem,
- Zastosowanie systemu CAPTCHA,
- Wymaganie jako hasła ciągu 8-16 znaków składających się ze znaków podstawowych (litery, cyfry), wyłączając znaki specjalne

4. Implementacja systemu baz danych

Implementacja bazy danych została wykonana w MySQL Workbench. Do łączenia się z bazą wykorzystujemy XAMPP.

4.1. Tworzenie tabel i definiowanie ograniczeń

Tabela admin

```
CREATE TABLE IF NOT EXISTS `sklep`.`admin` (
  `id_admin` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `admin_name` VARCHAR(45) CHARACTER SET 'utf8' COLLATE
'utf8_polish_ci' NOT NULL,
  `admin_nazwisko` VARCHAR(45) CHARACTER SET 'utf8' COLLATE
'utf8_polish_ci' NOT NULL,
  `admin_login` VARCHAR(45) CHARACTER SET 'utf8' COLLATE
'utf8_polish_ci' NOT NULL,
  `admin_password` VARCHAR(45) CHARACTER SET 'utf8' COLLATE
'utf8_polish_ci' NOT NULL,
  PRIMARY KEY (`id_admin`),
  INDEX `login` (`admin_login` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_polish_ci;
```

Tabela adresy

```
CREATE TABLE IF NOT EXISTS `sklep`.`adresy` (
  `id_adres` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `kod_pocztowy` INT UNSIGNED NOT NULL,
  `miejscowosc` VARCHAR(45) CHARACTER SET 'utf8' COLLATE
'utf8_polish_ci' NOT NULL,
  `ulica` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT
NULL,
  `numer_domu` INT NOT NULL,
  `numer_mieszkania` INT NULL,
  PRIMARY KEY (`id_adres`),
  INDEX `kod` (`kod_pocztowy` ASC) VISIBLE,
  CONSTRAINT `kod`
  FOREIGN KEY (`kod_pocztowy`)
  REFERENCES `sklep`.`kod_pocztowy` (`id_kod`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_polish_ci;
```

Tabela det_zam

```
CREATE TABLE IF NOT EXISTS `sklep`.`det_zam` (  
  `id_zam_pro` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nr_zamowienia` INT UNSIGNED NOT NULL,  
  `produkt` INT UNSIGNED NOT NULL,  
  `liczba` INT UNSIGNED NOT NULL,  
  `wartosc` FLOAT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id_zam_pro`),  
  INDEX `nr_zamow` (`nr_zamowienia` ASC) VISIBLE,  
  INDEX `nr_prod` (`produkt` ASC) VISIBLE,  
  CONSTRAINT `nr_zamow`  
    FOREIGN KEY (`nr_zamowienia`)  
    REFERENCES `sklep`.`zamowienia` (`id_zamowienia`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `nr_prod`  
    FOREIGN KEY (`produkt`)  
    REFERENCES `sklep`.`produkty` (`id_produkt`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_polish_ci;
```

Tabela kategorie

```
CREATE TABLE IF NOT EXISTS `sklep`.`kategorie` (  
  `id_kat` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nazwa` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT  
  NULL,  
  PRIMARY KEY (`id_kat`),  
  INDEX `nazwa` (`nazwa` ASC) VISIBLE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_polish_ci;
```

Tabela produkty


```

CREATE TABLE IF NOT EXISTS `sklep`.`produkty` (
  `id_produkt` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `kategoria` INT UNSIGNED NOT NULL,
  `nazwa` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT
  NULL,
  `producent` INT UNSIGNED NOT NULL,
  `cecha` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT
  NULL,
  `cena` FLOAT NOT NULL,
  `obraz` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci'
  NULL,
  `stan_magazynowy` INT UNSIGNED NOT NULL,
  `opis` LONGTEXT CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT
  NULL,
  PRIMARY KEY (`id_produkt`),
  INDEX `kategoria` (`kategoria` ASC) VISIBLE,
  INDEX `producent` (`producent` ASC) VISIBLE,
  INDEX `cecha` (`cecha` ASC) INVISIBLE,
  INDEX `cena` (`cena` ASC) INVISIBLE,
  INDEX `dostepnosc` (`stan_magazynowy` ASC) VISIBLE,
  CONSTRAINT `kategoria`
    FOREIGN KEY (`kategoria`)
      REFERENCES `sklep`.`kategorie` (`id_kat`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `producent`
    FOREIGN KEY (`producent`)
      REFERENCES `sklep`.`producenci` (`id_producent`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_polish_ci;

```

Tabela producenci

```

CREATE TABLE IF NOT EXISTS `sklep`.`producenci` (
  `id_producent` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `nazwa` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT
  NULL,
  PRIMARY KEY (`id_producent`),
  INDEX `nazwa` (`nazwa` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_polish_ci;

```

Tabela kod_pocztowy

```

CREATE TABLE IF NOT EXISTS `sklep`.`kod_pocztowy` (
  `id_kod` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `kod_pocztowy` VARCHAR(6) NOT NULL,
  `miasto` VARCHAR(45) NOT NULL,
  `wojewodztwo` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_kod`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_polish_ci;

```

Tabela users

```
CREATE TABLE IF NOT EXISTS `sklep`.`users` (  
  `id_user` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `user_imie` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci'  
    NOT NULL,  
  `user_nazwisko` VARCHAR(45) CHARACTER SET 'utf8' COLLATE  
    'utf8_polish_ci' NOT NULL,  
  `user_login` VARCHAR(45) CHARACTER SET 'utf8' COLLATE  
    'utf8_polish_ci' NOT NULL,  
  `user_password` VARCHAR(45) CHARACTER SET 'utf8' COLLATE  
    'utf8_polish_ci' NOT NULL,  
  `user_email` VARCHAR(45) CHARACTER SET 'utf8' COLLATE  
    'utf8_polish_ci' NOT NULL,  
  `user_adres` INT UNSIGNED NOT NULL,  
  `user_phone` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id_user`),  
  INDEX `adres` (`user_adres` ASC) INVISIBLE,  
  INDEX `imienazwisko` (`user_imie` ASC, `user_nazwisko` ASC) INVISIBLE,  
  INDEX `login` (`user_login` ASC) INVISIBLE,  
  CONSTRAINT `adres`  
    FOREIGN KEY (`user_adres`)  
    REFERENCES `sklep`.`adresy` (`id_adres`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_polish_ci;
```

Tabela zamowienia

```
CREATE TABLE IF NOT EXISTS `sklep`.`zamowienia` (  
  `id_zamowienia` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `klient` INT UNSIGNED NOT NULL,  
  `kiedy` DATE NOT NULL,  
  `kurier` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci'  
    NOT NULL,  
  `kwota` FLOAT UNSIGNED NOT NULL,  
  `status_zam` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id_zamowienia`),  
  INDEX `klient` (`klient` ASC) INVISIBLE,  
  INDEX `data_zam` (`kiedy` ASC) INVISIBLE,  
  INDEX `status` (`status_zam` ASC) VISIBLE,  
  CONSTRAINT `klient`  
    FOREIGN KEY (`klient`)  
    REFERENCES `sklep`.`users` (`id_user`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_polish_ci;
```

Tabela dostawa


```
CREATE TABLE IF NOT EXISTS `sklep`.`dostawa` (
  `id_dostawa` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `nazwa` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_polish_ci' NOT
  NULL,
  PRIMARY KEY (`id_dostawa`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_polish_ci;
```

4.2. Implementacja mechanizmów przetwarzania danych

- Widoki

Widok informacji o kliencie

```
CREATE TABLE IF NOT EXISTS `sklep`.`klient_info` (`imie` INT, `nazwisko`
  INT, `telefon` INT, `email` INT, `kod_pocztowy` INT, `miasto` INT, `
  wojewodztwo` INT, `miejscowosc` INT, `ulica` INT, `nr_domu` INT, `
  nr_mieszkania` INT);
```

widok wszystkich kategorii

```
CREATE TABLE IF NOT EXISTS `sklep`.`kategorie_widok` (`kategoria` INT);
```

widok do logowania administratora

```
CREATE TABLE IF NOT EXISTS `sklep`.`log_admin` (`login` INT, `haslo`
  INT);
```

widok do logowania klienta

```
CREATE TABLE IF NOT EXISTS `sklep`.`log_klient` (`login` INT, `haslo`
  INT);
```

widok przedmiotów w sklepie

```
CREATE TABLE IF NOT EXISTS `sklep`.`towar` (`id` INT, `kategoria` INT,
  `nazwa` INT, `producent` INT, `cena` INT, `cecha` INT, `obraz` INT, `
  dostepnosc` INT, `opis` INT);
```

widok zamówienia

```
CREATE TABLE IF NOT EXISTS `sklep`.`zamowienie_widok` (`id_klienta` INT
  , `id_zamowienia` INT, `data_zamowienia` INT, `kurier` INT, `wartosc`
  INT, `status_zamowienia` INT);
```

Widok prod_w_zamowieniu

```
CREATE TABLE IF NOT EXISTS `sklep`.`prod_w_zamowieniu` (`id_zamowienia` INT
  , `id_produkta` INT, `liczba` INT, `wartosc` FLOAT);
```

- Procedury
 - dodaj_klienta(imie,nazwisko,login,haslo,telefon,email,kod_pocztowy,miasto,wojewodztwo,miejscowosc,ulica,nr_domu,nr_mieszkania)

Procedura z dodaje klienta, jego adres oraz kod_pocztowy o ile nie istnieją

- edytuj_klienta(imie,nazwisko,telefon,email,kod_pocztowy,miasto,województwo,miejsco
wosc,ulica,nr_domu,nr_mieszkania)
- dodaj_produkt(kategoria,nazwa,producent,cena,cecha,obraz,dostepnosc,opis)
- edytuj_produkt(kategoria,nazwa,producent,cena,cecha,obraz,dostepnosc,opis)
- dodaj_kategorie(kategoria)
- edytuj_kategorie(kategoria)
- dodaj_producenta(nazwa)
- edytuj_producenta(nazwa)
- dodaj_zamowienie(klient,kiedy,dostawca,kwota,status)
- dodaj_produkt_do_zamowienia(zamowienie,produkt,liczba,wartosc_zamowienia)
- edytuj_status(status)

4.3. Implementacja uprawnień i innych zabezpieczeń

W celu ochrony danych z bazy danych zostały stworzone trzy poziomy uprawnnień. Użytkownicy serwisu mają ograniczony dostęp do danych, dzięki czemu baza danych zyskuje znacznie na bezpieczeństwie. Nadawanie uprawnień jest zdefiniowane za pomocą ról, które w prosty sposób pozwalają grupować użytkowników na trzy docelowe grupy.

W naszej bazie zaimplementowane zostały poziomy uprawnnień:

- administrator
- klient
- gość

Każda tabela, widok czy procedura ma docelowych użytkowników, którzy mają dostęp do jej wnętrza. Po stworzeniu klienta możemy w łatwy sposób nadać mu jego rolę.

4.4. Testowanie bazy danych na przykładowych danych

- Test 1. dodaj_klienta

```
CALL dodaj_klienta("Marek", "Konrad", "BZWBK", "BZWBKmoimzyciem",  
6666666666, "m.konrad@wp.pl", "99-100", "Wrocław", "Dolnośląskie",  
"Wrocław", "Zakopiańska", 11, 2)
```

Widok klient_info

imie	nazwisko	telefon	email	kod_poc	miasto	województwo	miescowosc	ulica	nr_domu	nr_mieszkania
Marek	Konrad	6666666666	m.konrad@wp.pl	99-100	Wrocław	Dolnośląskie	Wrocław	Zakopiańska	11	2

Informacje o kliencie zostały dodane do odpowiednich tabel a wynikiem tego jest pojawienie się nowego klienta w widoku klient_info.

- Test 2. dodaj_produkt

```
CALL dodaj_produkt("Rezystory", "Kolonafida THT", "Philips", 12.99, "12M Ohm",
"img/kolonafidatht.png", 10000, "Nowość! Pierwszy tranzystor Philipsa na
rynku. Największa moc, największa rezystancja. Sprawdź to sam!")
```

Widok towar

id	kategoria	nazwa	producent	cena	cecha	obraz	dostepnosc	opis
1	Rezystory	Kolonafida THT	Philips	12.99	12M Ohm	img/kolonafidatht.png	10000	Nowość! Pierwszy tranzystor Philipsa na rynku. Naj...

Informacje o produkcie zostały odpowiednio dodane do tabel, czego wynikiem jest ukazanie się nowego rekordu w widoku towar.

- Test 3. dodaj_zamowienie oraz dodaj_produkt_do_zamowienia

```
CALL dodaj_zamowienie(1, STR_TO_DATE('21/05/2019', '%d/%m/%y'), 2, 38.97, 0)
CALL dodaj_produkt_do_zamowienia(1, 1, 3, 38.97)
```

Widok zamowienia

id_klienta	id_zamowienia	data_zamowienia	kurier	wartosc	status_zamowienia
1	1	2019-05-21	UPS	38.97	0

Widok prod_w_zamowieniu

id_zamowienia	id_produkt	liczba	wartosc
1	1	3	38.97

Informacje o zamówieniu oraz o produkcie zostały dodane do odpowiednich tabel o czym możemy się przekonać w widoku zamowienia oraz prod_w_zamowieniu.

- Test 4. Zabezpieczenia

Po próbie wywołania procedury dodającej produkt z konta klienta otrzymaliśmy następujący komunikat:

```
Execute command denied to user 'klient'@'%' for routine `sklep.
dodaj_produkt`
```

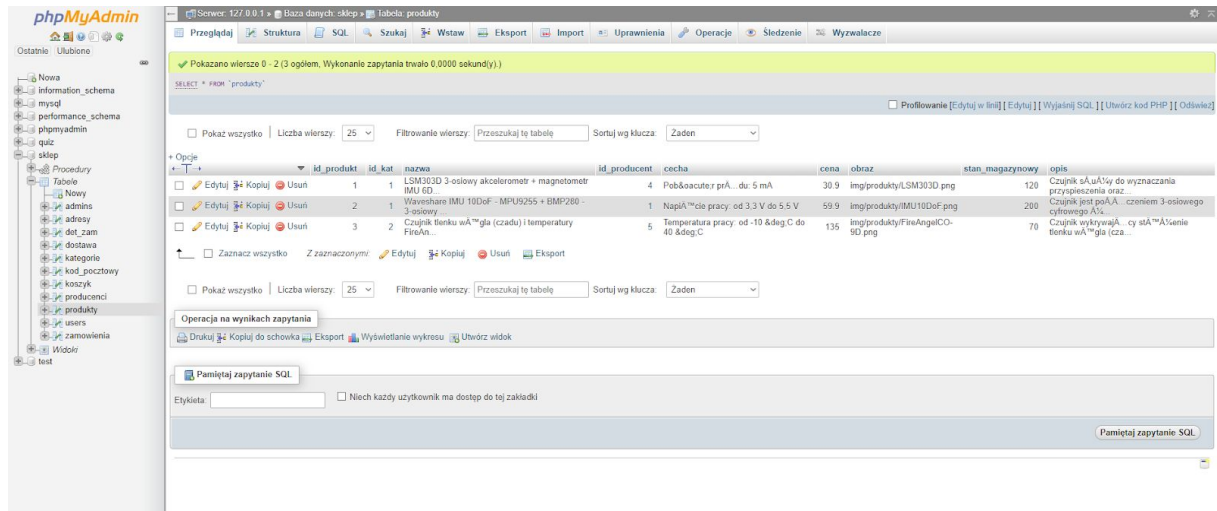
Dokładnie takiej reakcji ze strony bazy się spodziewaliśmy, gdyż klient nie ma uprawnień do dodawania produktów. Dzięki takiemu wynikowi wiemy, iż nadawanie uprawnień działa i tylko osoby uprawnione mają dostęp do odpowiednich procedur.

5. Implementacja i testy aplikacji

Aplikacja dostępowa została napisana przy wykorzystaniu: PHP, HTML oraz CSS. Jako, że to była nasza pierwsza styczność z modelowaniem stron internetowych postawiliśmy na sprawdzone technologie z mnogością poradników. Część naszej grupy już wcześniej miała styczność z PHP co ułatwiło proces tworzenia aplikacji.

5.1. Implementacja interfejsu dostępu do bazy danych

Jako interfejs dostępu do bazy danych wybraliśmy phpmyadmin. Przy wyborze kierowaliśmy się prostotą, a także dostępnością tutoriali w internecie. Cały proces zarządzania naszymi użytkownikami, produktami odbywa się przy wykorzystaniu tego panelu.



5.2. Implementacja wybranych funkcjonalności

Implementacja funkcji wyświetlania listy produktów polegała na połączeniu oraz odczytaniu danych z bazy SQL, następnie pobrane wyniki były prezentowane na stronie w odpowiednich kategoriach z odpowiednimi opisami oraz obrazkami.

Przykładowe połączenie z bazą (wykorzystując funkcję mysqli) oraz zapytanie do bazy:

```
$polaczenie = @new mysqli($host,$db_user,$db_password,$db_name);  
$zapytanie = "SELECT * FROM towar";  
$rezultat = mysqli_query($polaczenie, $zapytanie);  
$ile = mysqli_num_rows($rezultat);
```

Przy czym odpowiednio:

- host - numer IP hosta
- db_user- nazwa użytkownika
- db_password - hasło użytkownika
- db_name - nazwa bazy do której przesyłane jest zapytanie

Następnie wyświetlenie otrzymanych rezultatów realizuje funkcja `mysqli_fetch_assoc()`.

```
$row = mysqli_fetch_assoc($rezultat);  
$id = $row['id'];  
$kat = $row['kategoria'];  
$nazwa = $row['nazwa'];  
$producent = $row['producent'];  
$cecha = $row['cecha'];  
$cena = $row['cena'];  
$obraz = $row['obraz'];  
$stan_magazynowy = $row['dostepnosc'];  
$opis = $row['opis'];
```

Wyświetlanie oraz edycja produktów na stronie polegała na przekazywaniu zmiennych pomiędzy skryptami. Oglądanie produktów na stronie jest dostępne dla każdego, po zalogowaniu jest możliwość przejrzania swoich zamówień.

5.3. Implementacja mechanizmów bezpieczeństwa

Podstawowym kryterium mechanizmu bezpieczeństwa jest zgodność loginu i hasła do danych zawartych w bazie. Jeśli wprowadzone zostaną złe dane próba dostępu zostanie zakończona stosownym komunikatem. Przy dostępie do każdej podstrony jak i strony głównej system dba o to, czy przeglądający jest zalogowanym użytkownikiem czy gościem w ten sposób widoczne są dodatkowe funkcjonalności przykładowo historia zamówień. Kolejnym mechanizmem jest szyfrowanie haseł użytkowników:

```
$haslo_hash = password_hash($haslo, PASSWORD_DEFAULT);
```

gdzie "hasło" jest podanym hasłem, a PASSWORD_DEFAULT metodą szyfrowania.

6. Instrukcja użytkownika strony

Strona główna posiada różne zakładki. Aby przejść do żądanej zakładki należy najechać na nią kursorem myszy i kliknąć lewy przycisk myszy. Uzyskiwanie praw klienta odbywa się poprzez wypełnienie odpowiedniego formularza w zakładce "zarejestruj się".

Strona administratorska jest dostępna poprzez specjalny link.

7. Podsumowanie

Projekt wraz z poszczególnymi etapami ulegał modyfikacjom. Finalnie udało się zachować początkowy zamysł, natomiast wykorzystywane technologie uległy zmianom.

Aplikacja pomimo aktualnie niskiej złożoności, ma różne możliwe drogi rozwoju. Kolejnym głównym celem jest zaimplementowanie dedykowanego systemu obsługi płatności, poza tym można też stworzyć algorytm podpowiadający produkty w zależności od wcześniejszych zakupów klienta. Warto też ulepszyć aspekt estetyczny aplikacji.

W projekcie udało nam się zaznajomić z wykorzystanymi technologiami. Poszerzyliśmy swoje zdolności pracy zespołowej, a także co najważniejsze udało nam się stworzyć w pełni funkcjonalny serwis przy początkowym, słabym, zaznajomieniu z tematem.