

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

PROJEKT Z BAZ DANYCH

Projekt bazy danych sklepu komputerowego

Termin zajęć: środa 11:15

Prowadzący zajęcia: dr inż. Roman Ptak, W4/K9

WROCŁAW 2019

Spis treści:

1. Wstęp	3
1.1. Cel projektu	3
1.2. Zakres projektu	3
2. Analiza wymagań	4
2.1. Opis działania i przyjęte założenia projektowe	4
2.2. Wymagania funkcjonalne	4
2.3. Wymagania niefunkcjonalne	5
3. Projekt systemu	6
3.1. Analiza rzeczywistości i uproszczony model konceptualny	6
3.2. Model fizyczno-logiczny i normalizacja	7
3.3. Schematy UML	8
3.4. Konfiguracja środowiska	13
3.5. Projekt mechanizmów bezpieczeństwa	13
3.6. Projekt mock-up'ów	15
4. Implementacja systemu baz danych	18
4.1. Implementacja systemu baz danych	18
4.2. Testowanie bazy danych na przykładowych danych	25
5. Implementacja i testy aplikacji	38
5.1. Instalacja i konfigurowanie systemu	38
5.2. Instrukcja użytkowania aplikacji	49
5.3. Testowanie opracowanych funkcji systemu	57
5.4. Omówienie wybranych rozwiązań programistycznych	58
5.4.1. Implementacja interfejsu dostępu do bazy danych	58
5.4.2. Implementacja wybranych funkcjonalności systemu	58
5.4.3. Implementacja mechanizmów bezpieczeństwa	62
6. Podsumowanie i wnioski	62
Literatura	64
Spis rysunków	65
Spis tabel	67

1. Wstęp

1.1. Cel projektu

Celem projektu jest zaprogramowanie i stworzenie systemu baz danych wraz z interfejsem internetowym. W szczególności celem pracy jest:

- opracowanie założeń dotyczących bazy,
- opracowanie systemu baz danych,
- opracowanie aplikacji dostępowej,
- opracowanie dokumentacji serwisowej.

1.2. Zakres pracy

Zakres pracy zawiera zaprogramowanie systemu baz danych. Należy także zaprogramować internetową aplikację dostępową. W szczególności zakres pracy obejmuje:

- Należy sformułować założenia projektowe oraz wymagania funkcjonalne oraz нефункциональные. Należy przygotować dokumentację omawiającą te zagadnienia.
- Należy zaprojektować schematy UML oraz schematy fizyczno-logiczne. Należy skonstruować dokumentację opisującą ich logikę.
- Należy stworzyć bazę danych korzystając z środowiska MySQL. Należy uruchomić ją korzystając z PHPMysqlAdmin na lokalnym urządzeniu. Należy przygotować dokumentację opisującą ten fragment projektu. Należy skonstruować przykładowe mock-up'y aplikacji dostępowej.
- Należy opracować aplikację dostępową korzystając z PHP. Należy wykonać dokumentację by wskazać kierunek konstruowania podobnej bazy przez inną osobę.

2. Analiza wymagań

2.1. Opis i przyjęte założenia projektowe

Nasza firma będzie zajmowała się sprzedażą internetową części, akcesoriów, podzespołów komputerowych oraz składaniem ich w gotowe zestawy komputerowe. Będzie to mały biznes w rodzinnym mieście nieposiadający fizycznego sklepu, a wszystkie części, które będzie można zakupić na stronie będą przetwarzane wprost z magazynu. Podział na użytkowników będzie następujący: klient oraz administrator. Ewentualne przechowywanie produktów przed wysłaniem całości lub w celu zmontowania całego zestawu w prywatnym mieszkaniu.

2.2. Wymagania funkcjonalne:

➤ Klient:

- Kupowanie produktów na naszej stronie – zarówno pojedynczych części jak i gotowych zestawów. Za dodatkową opłatą istnieje możliwość złożenia całego komputera z wybranych uprzednio części.
- W celu ułatwienia użytkownikowi procesu złożenia całego komputera istnieje możliwość wykorzystania konfiguratora, który będzie informował jakich części potrzebujemy, aby skompletować zestaw.
- Rejestracja konta poprzez podanie danych takich jak: nazwa użytkownika, hasło oraz e-mail z możliwością rejestracji zarówno firm jak i osób.
- Możliwość wyszukiwania produktów.
- Możliwość sprawdzenia czy pożądaný produkt jest dostępny w magazynie.
- Wkładanie interesujących nas produktów do koszyka sklepowego, który potem umożliwia ich podejrzenie przed sfinalizowaniem transakcji.
- Możliwość dokonania wyboru sposobu płatności.
- Po dokonaniu transakcji użytkownik dostaje potwierdzenie zakupu na swój adres e-mail podany podczas rejestracji.
- Odzyskiwanie hasła.
- Możliwość zgłoszenia reklamacji wcześniej zakupionego produktu.

➤ Administrator:

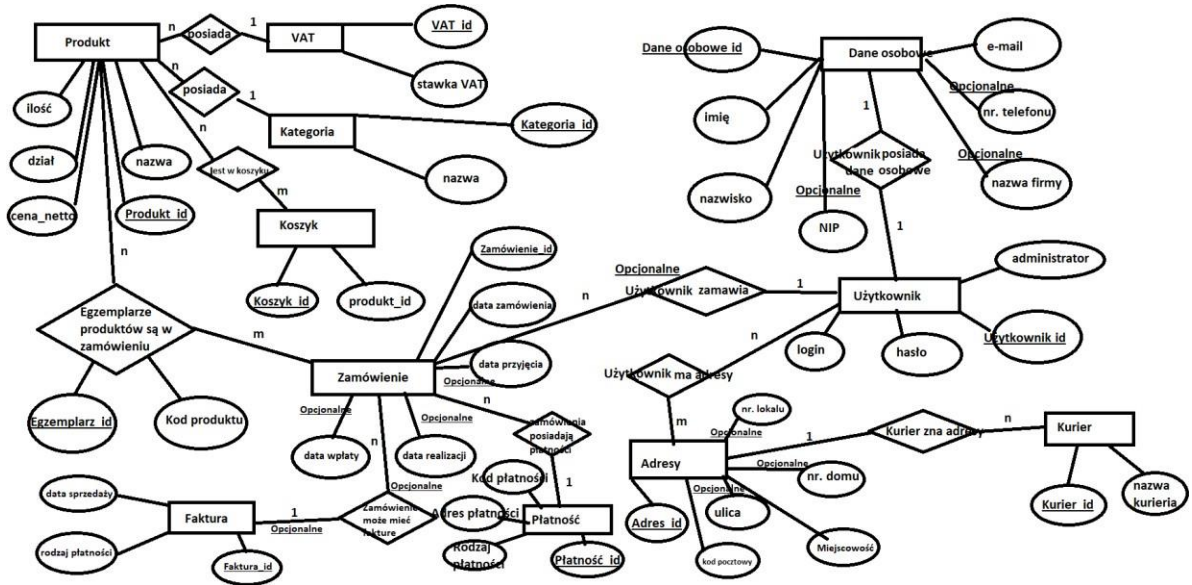
- Sprawdzanie stanu produktów w magazynie.
- Możliwość dodawania produktów znajdujących się w bazie danych.
- Możliwość edycji produktów znajdujących się w bazie danych.
- Możliwość usuwania produktów z bazy danych.
- Możliwość zamawiania produktów do magazynu.
- Możliwość usuwania i dodawania dostawców do bazy.
- Resetowanie haseł użytkowników.
- Wyszukiwanie firm po numerze NIP.
- Wystawianie faktur oraz paragonów.
- Dostęp do danych klienta takich jak adres zamieszkania, numer telefonu w celu przekazania ich dalej kurierowi.
- Z racji tego, iż nasza firma nie posiada własnego kuriera administrator ma możliwość zamówienia kuriera z firm zewnętrznych.

2.3. Wymagania niefunkcjonalne:

- Ze względu na zawarte w systemie dane osobowe, system będzie wymagał zalogowania się w razie nieautoryzowanej próby dostępu do nich
- Autoryzowanie użytkownika za pomocą tradycyjnego systemu loginu i hasła
- W razie wielu nieudanych prób logowania konieczność wpisania kodu captch'a by uniknąć prób siłowego łamania haseł
- Hasła użytkownika jak i administratora będą szyfrowane
- Do wykonania modelu koncepcyjnego bazy zostanie wykorzystany MySQL Workbench
- Serwer zostanie utworzony przy pomocy MySQL
- Aplikacja dostępowa będzie przeznaczona do otwierania głównie na komputerach stacjonarnych, ewentualnie urządzeniach mobilnych
- Przewidujemy, że liczba naszych klientów będzie wynosiła około 100 osób. Każdy będzie wykonywał średnio 5 transakcji rocznie co daje nam około 500 rekordów w naszej bazie.
- Jako przypadek ekstremalny zakładamy obsługę do 100 klientów naraz.

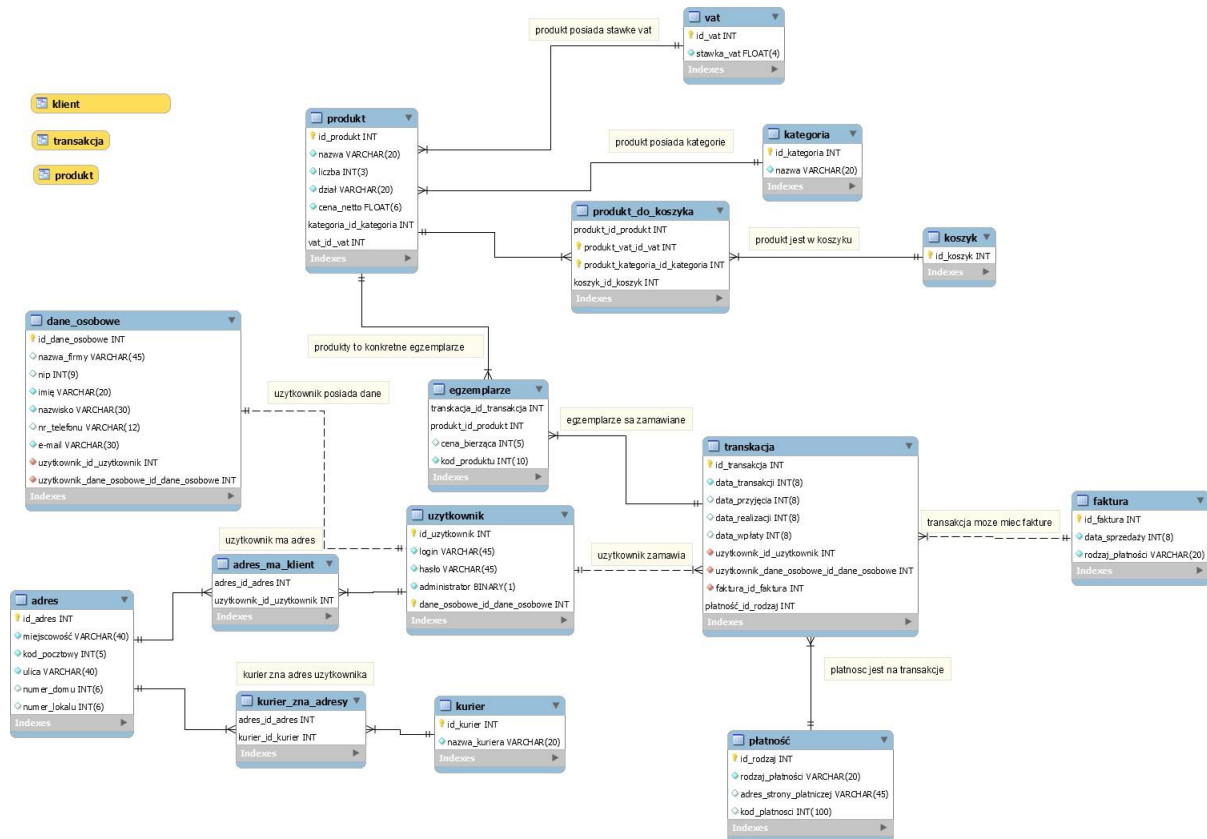
3. Projekt systemu

3.1. Uproszczony model konceptualny



Rysunek 1 Model konceptualny

3.2. Model logiczno-fizyczny i normalizacja



Rysunek 2: Model logiczno-fizyczny

klient

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

1 • CREATE VIEW `klient` AS SELECT *
2   FROM `dane_osobowe` CROSS JOIN `uzytkownik` `adres`

```

Rysunek 3: Widok produktu

transakcja

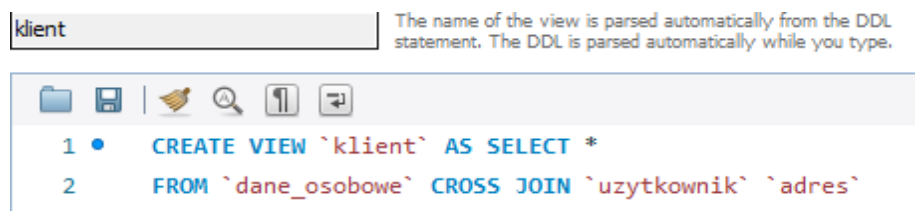
The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

1 • CREATE VIEW `transakcja` AS SELECT *
2   FROM `faktura` CROSS JOIN `platnosc` `transakcja`

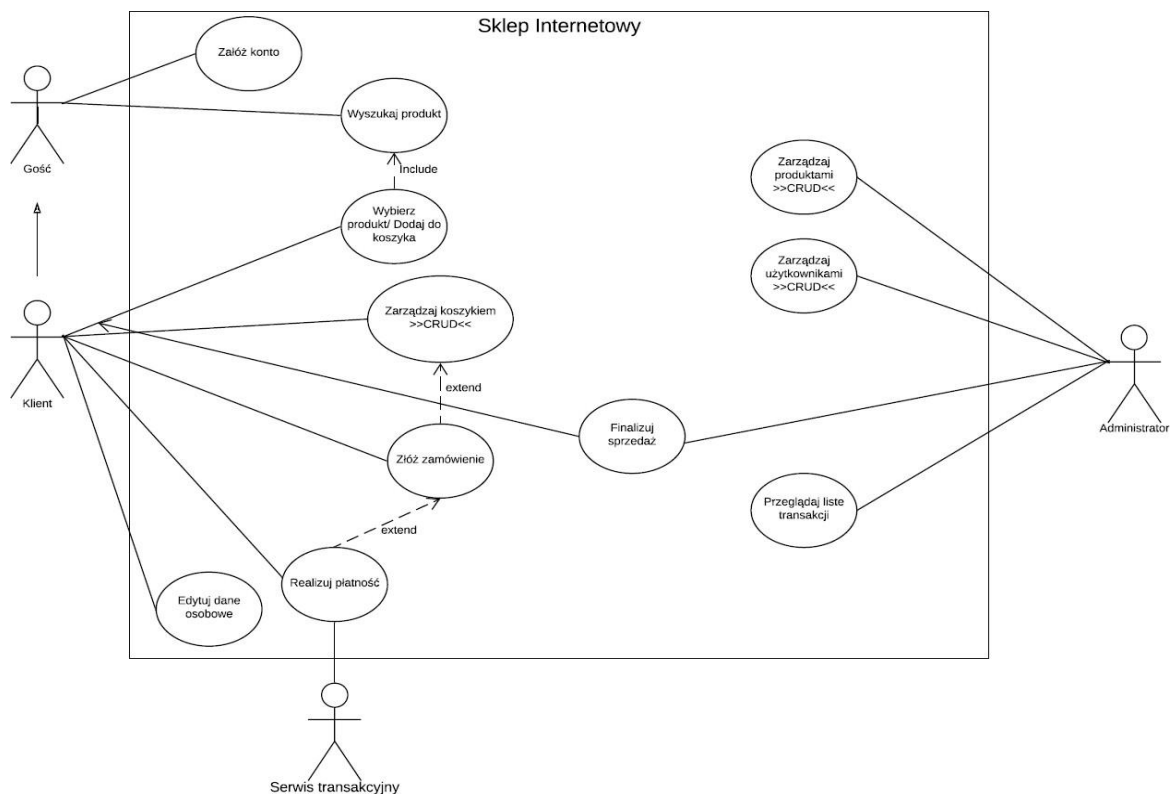
```

Rysunek 4: Widok transakcji

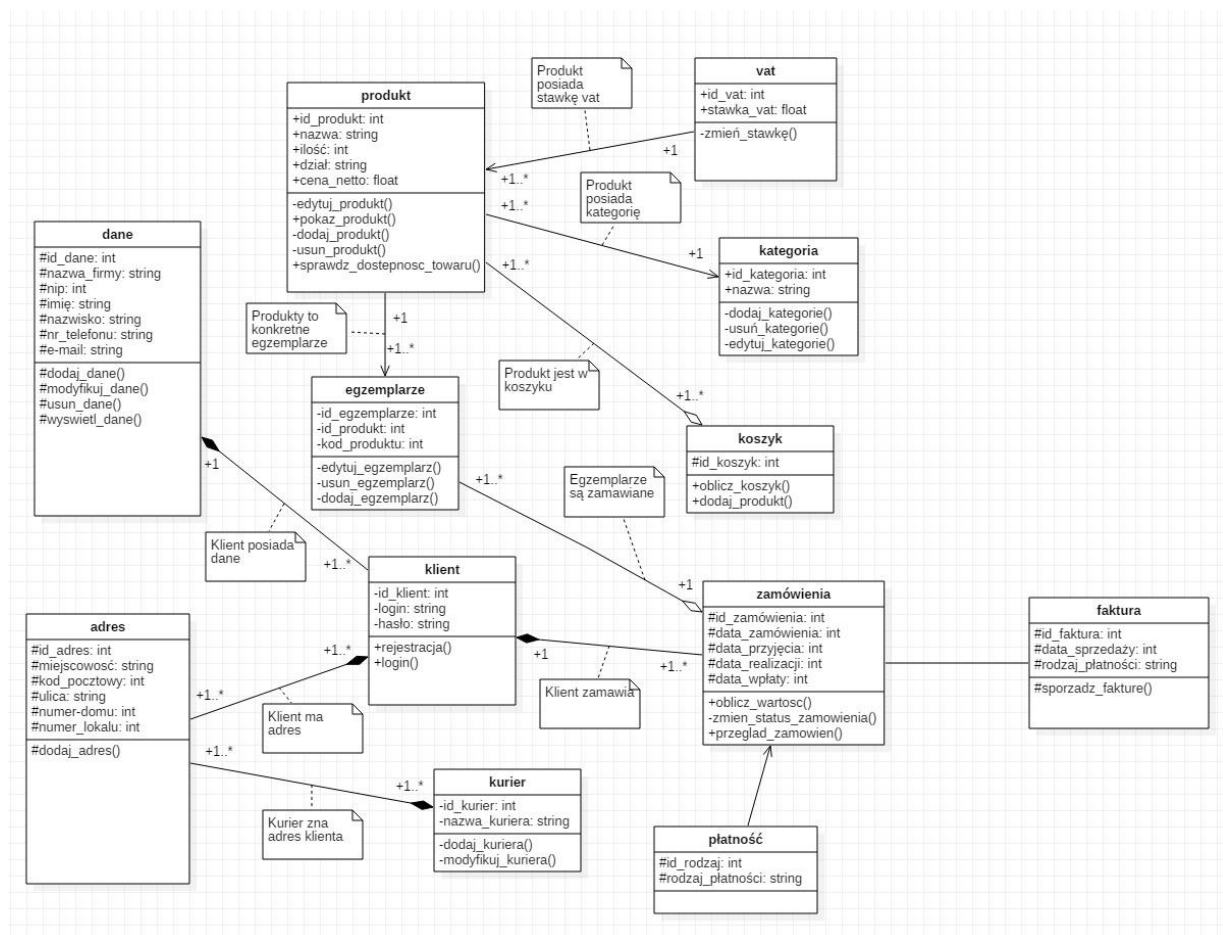


Rysunek 5: Widok klienta

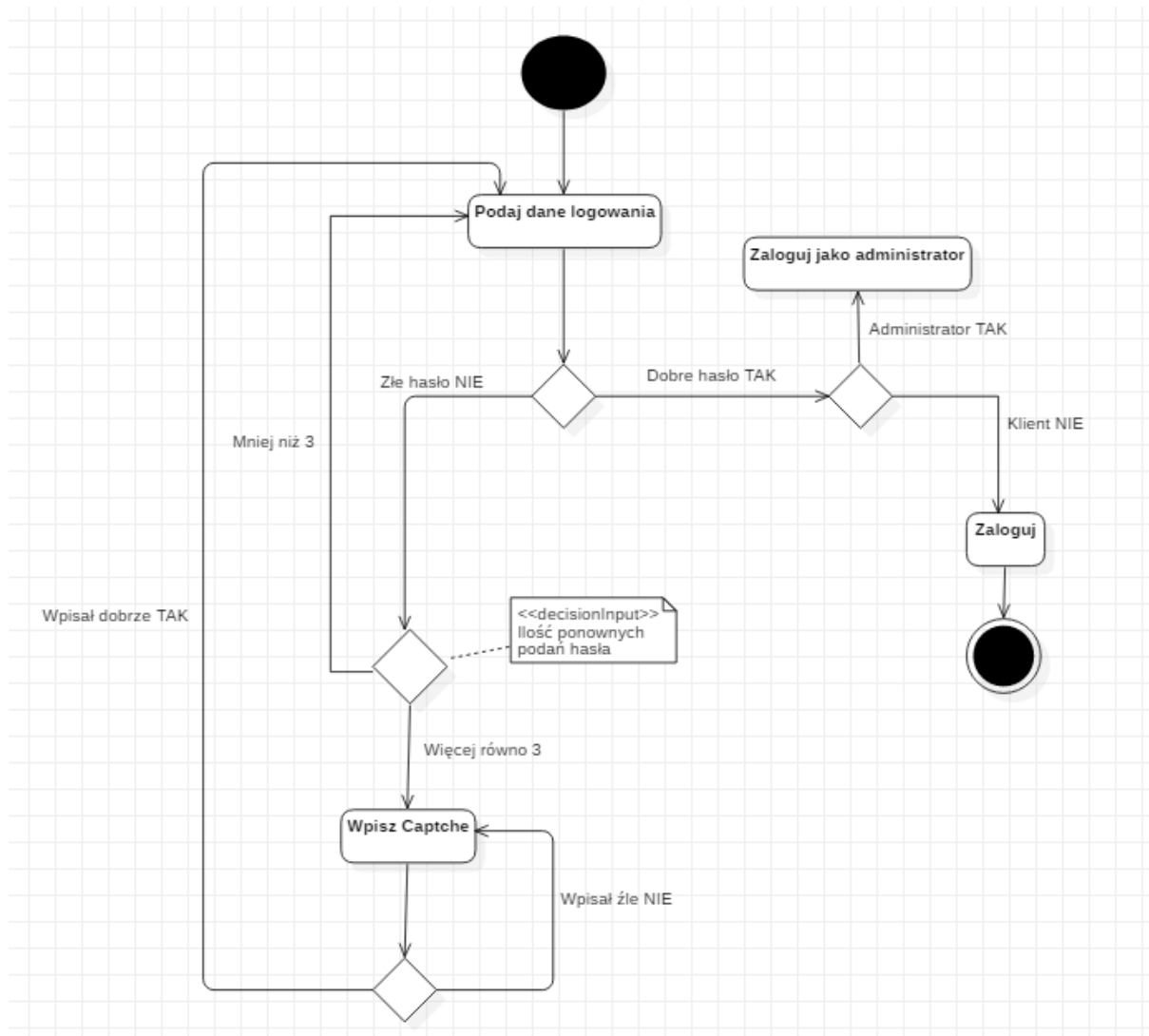
3.3. Schematy UML



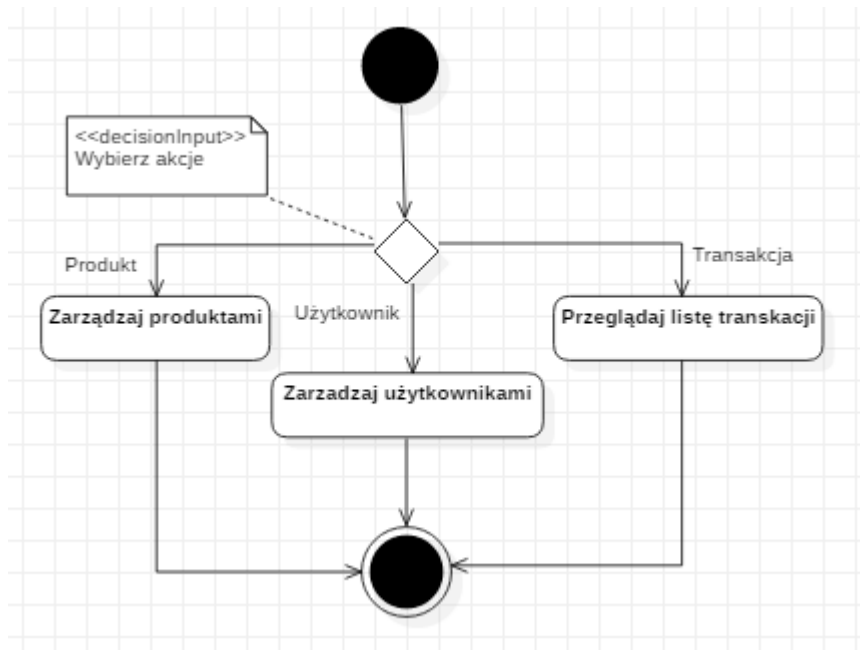
Rysunek 6: Diagram przypadków użycia UML



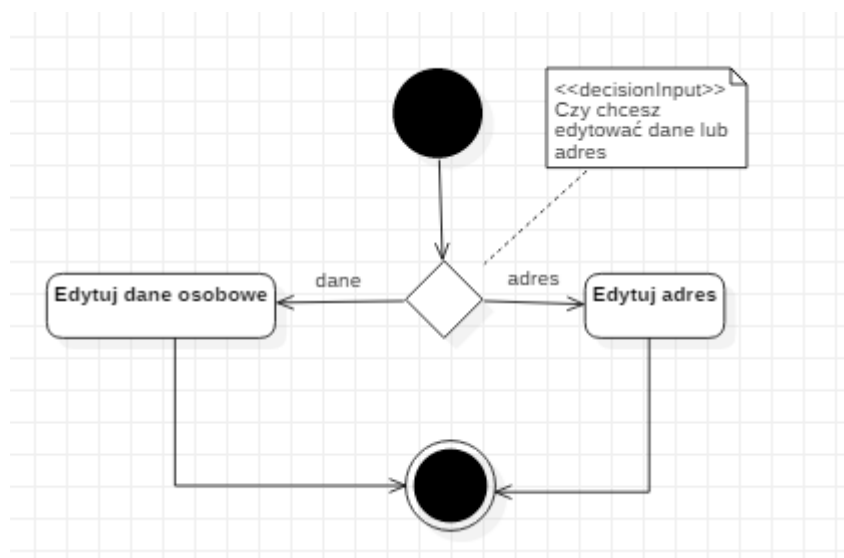
Rysunek 7: Diagram klas UML



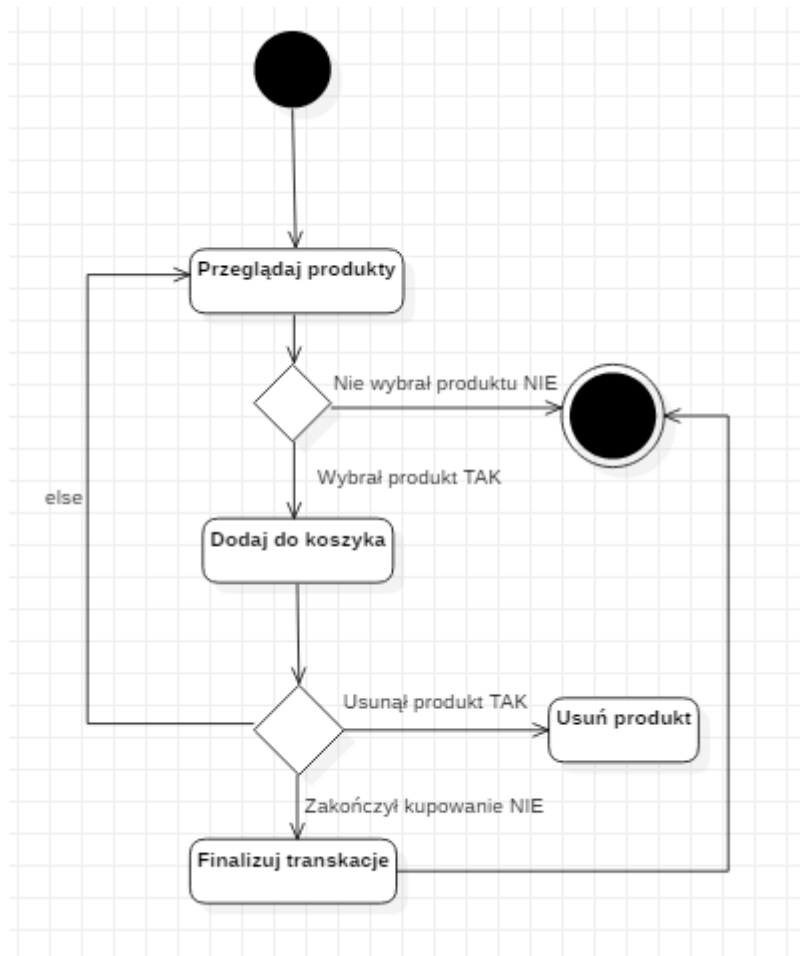
Rysunek 8: Diagram czynności UML - logowanie



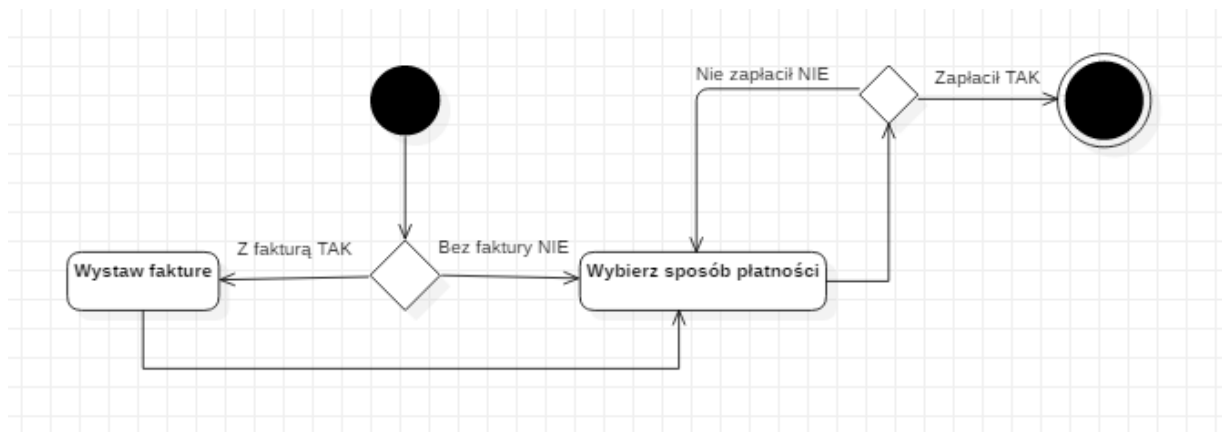
Rysunek 9: Diagram czynności UML - administrator



Rysunek 10: Diagram czynności UML - edycja danych osobowych lub adresu



Rysunek 11: Diagram czynności UML - przeglądanie produktów



Rysunek 12: Diagram czynności UML - płatność

3.4. Konfiguracja środowiskowa

Zaczynamy od pobrania ze strony dewelopera MySQL server. Następnie przystępujemy do konfiguracji naszego serwera MySQL poprzez wybranie nazwy serwera, metody połączenia oraz portu. Typ połączenie, nazwę oraz port zostawiliśmy domyślne, czyli TCP/IP, MYSQL oraz 3306. Potem należy utworzyć użytkownika root oraz nadać mu silne hasło. Po wykonaniu powyższych czynności postawiliśmy nasz pierwszy serwer MySQL na komputerze.

Następnym krokiem jest wybranie connector'a, który pozwoli połączyć naszą bazę danych z wybraną przez nas aplikacją do zarządzania bazą danych. Tutaj wybór padł na ODBC driver firmy Microsoft, a dokładniej to MySQL ODBC 8.0 Driver.

Do zarządzania naszą wybraliśmy program MySQL Workbench, który posłużył nam do utworzenia bazy danych ze schematu fizycznego. Wykorzystaliśmy do tego funkcję Forward Engineer, która połączyła się ze wcześniej postawionym serwerem MySQL i utworzyła bazę danych wykorzystując do tego silnik InnoDB w wersji 8.0.

3.5. Projekt mechanizmów bezpieczeństwa

Baza danych:

- Dane będą szyfrowane – ciąg znaków będzie zamieniany na ciąg cyfr według klucza prywatnego
- Autoryzacja użytkownika oraz administratora jako login i hasło, które są szyfrowane

Klasa	Administrator	Klient	Gość	Serwis transakcyjny
Produkt	All	R	R	-
Egzemplarz	All	-	-	-
VAT	All	R	R	-
Dane osobowe	All	CRUD	-	-
Adres	All	CRUD	-	-
Użytkownik	All	RU	-	-
Kurier	All	-	-	-
Koszyk	All	RUD	-	-
Transakcja	All	CRU	-	-
Płatność	All	CRU	-	RU
Faktura	All	R	-	-
Kategoria	All	R	R	-

1 Tabela uprawnień

Umiejscowienie bazy danych:

- Sieć VPN
- Tunelowanie z wykorzystaniem SSL

Składowanie:

- Dane będą składowane w naszym domu
- Jako że posiadamy części komputerowe jako sklep zawsze mamy redundancję sprzętową zapewnioną
- W razie braku prądu posiadamy agregat prądowy
- Posiadamy alarm w domu

Błędy w oprogramowaniu

- Macierz RAID 5 dysków
- Ochrona antywirusowa
- Aktualizacje
- Zapora sieciowa
- Projektowanie, implementacja, testowanie i walidacja oprogramowania metodami formalnymi
- Backupy pełne - sieciowy i lokalny

Uwierzytelnianie i autoryzacja

- Grupy – klient i administrator
- Monitorowanie i rejestrowanie zdarzeń na serwerze w poszukiwaniu nietypowych zdarzeń
- Szyfrowanie z kluczem publicznym

SQL Injection:

- Niedopuszczenie do przekazania znaku ' do zapytania
- Wyłączenie komunikatów o błędach
- Rzutowanie łańcucha znaków na wartość liczbową
- Ograniczenia uprawnień

3.6. Projekt mock-up'ów

A registration form titled "REJESTRACJA" with three input fields: "Login", "Hasło", and "Adres e-mail". Below the fields is a button labeled "REJESTRUJ".

REJESTRACJA	
Login	
Hasło	
Adres e-mail	
<input type="button" value="REJESTRUJ"/>	

Rysunek 13: Mock-up rejestracji

A form titled "DANE" for editing personal data. It contains several input fields: "Imię", "Nazwisko", "Adres e-mail", "Telefon", "Miejscowość", "Kod pocztowy", "Ulica", "Nr domu", "Nr lokalu", "Konto firmowe" (with a checked checkbox), "NIP", and "Nazwa firmy". At the bottom are two buttons: "COFNIJ" and "ZAPISZ".

DANE		
Imię	Nazwisko	
Adres e-mail	Telefon	
Miejscowość	Kod pocztowy	
Ulica	Nr domu	Nr lokalu
<input checked="" type="checkbox"/> Konto firmowe		
NIP	Nazwa firmy	
<input type="button" value="COFNIJ"/> <input type="button" value="ZAPISZ"/>		

Rysunek 14: Mock-up edycji danych osobowych



Rysunek 15: Mock-up wyszukiwania produktu



Rysunek 16: Mock-up produktu

Koszyk

		Cena	Ilość	
	<div></div>	29.99 zł	1 <div></div>	<div>usuń</div>
	<div></div>	40.00 zł	2 <div></div>	<div>usuń</div>
	<div></div>	20.00 zł	3 <div></div>	<div>usuń</div>
	<div></div>	40.00 zł	2 <div></div>	<div>usuń</div>
	<div></div>	20.00 zł	3 <div></div>	<div>usuń</div>

Do zapłaty: 1445 zł

Kontynuuj zakupy

Złóż zamówienie

Rysunek 17: Mock-up koszyka

Płatność i wybór adresu wysyłki

Metody płatności:

☐ Karta kredytowa



CVV:

☐ Przelew internetowy



Adres wysyłki:

☐ Adres podany w danych konta

☐ Inny

KONTYNUUJ

Rysunek 18: Mock-up płatności

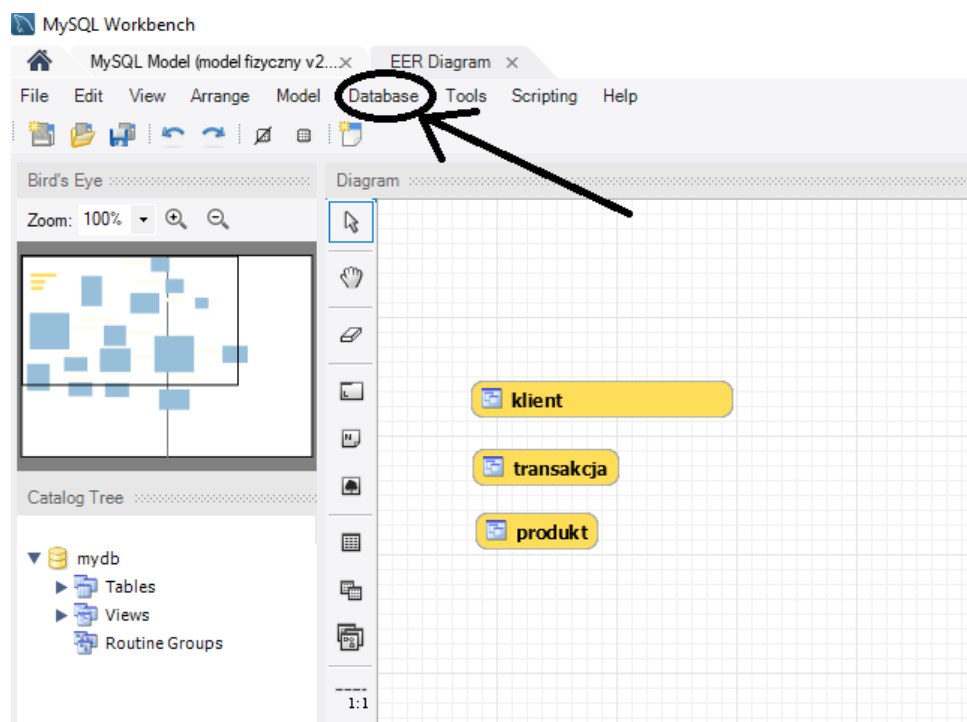
4. Implementacja systemu baz danych

4.1. Implementacja systemu baz danych

Implementacja została wykonana przy pomocy wcześniej wybranego systemu do zarządzania bazami danych – MySQL Workbench. Na samym początku postawiliśmy lokalny serwer MySQL dla naszej bazy danych. Cała konfiguracja została opisana w punkcie dotyczącym konfiguracji środowiska.

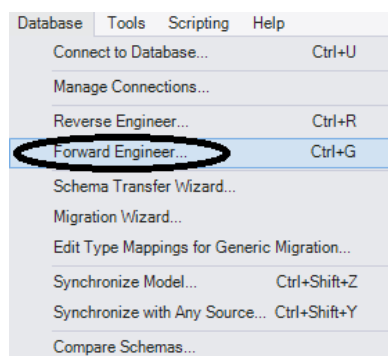
Następnie wygenerowaliśmy bazę korzystając z naszego modelu fizycznego. Z racji tego, że model posiada zdefiniowane, poprzez relacje ograniczenia oraz zabezpieczenia nie musieliśmy implementować ich na nowo korzystając z skryptów SQLowych.

Aby tego dokonać należało wybrać opcję Database znajdującą się na górnym pasku w lewym rogu okna programu.



Rysunek 19: Implementacja bazy danych krok 1

Po otwarciu się okienka opcji do wyboru należało kliknąć Forward Engineer (można również wykorzystać skrót klawiszowy Ctrl+G).

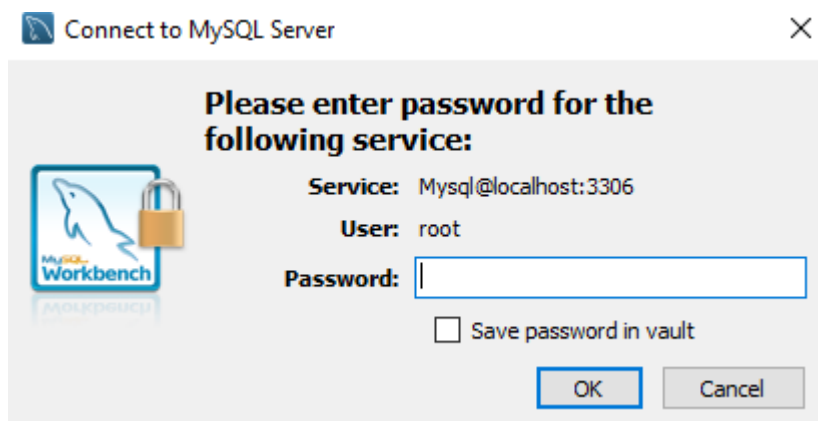


Rysunek 20: Implementacja bazy danych krok 2 - korzystanie z funkcji forward engineer

Następnie wypełniamy dane zgodnie z tymi, które podawaliśmy przy stawianiu serwera, dotyczące Hosta, portu oraz nazwy użytkownika, potem klikamy Next.

Rysunek 21: Implementacja bazy danych krok 3 - wybór parametrów

Później wyskakuje okienko z opcjami, tutaj zostawiłem wszystko domyślnie i kliknąłem Next. Następnie program poprosi nas o podanie hasła dla roota naszego serwera:



Rysunek 22: Implementacja bazy danych krok 4 - logowanie się jako root serwera MySQL

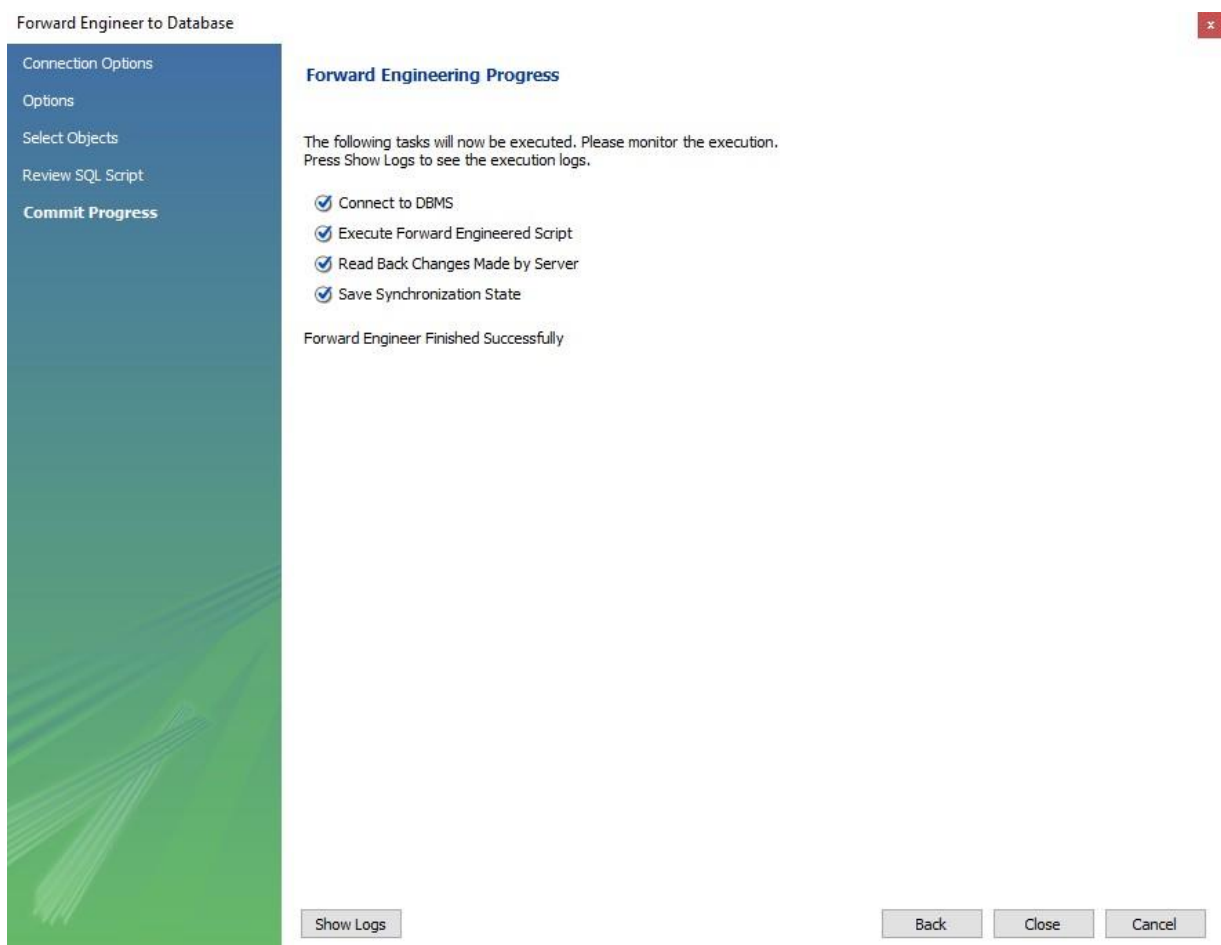
Wypełniamy hasło i klikamy OK.

Zaznaczamy kolejno, aby wyeksportować tabele wraz z utworzonymi widokami i klikamy Next:



Rysunek 23: Implementacja bazy danych krok 5 - wybór elementów do eksportu

Po zaakceptowaniu SQLowego skryptu i przejściu nadal nastąpi implementacja naszej bazy. Jeśli wszystko pójdzie bez problemu otrzymamy następujące okienko:



Rysunek 24: Komunikat o poprawnej implementacji bazy danych

Fragmenty kodu wygenerowanego przy użyciu funkcji Forward Engineer:

```
10
11  -----
12  -- Schema mydb
13  -----
14  CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
15  USE `mydb` ;
16
17  -----
18  -- Table `mydb`.`adres`
19  -----
20  CREATE TABLE IF NOT EXISTS `mydb`.`adres` (
21    `id_adres` INT NOT NULL,
22    `miejscowość` VARCHAR(40) NOT NULL,
23    `kod_pocztowy` INT(5) NOT NULL,
24    `ulica` VARCHAR(40) NOT NULL,
25    `numer_domu` INT(6) NULL,
26    `numer_lokalu` INT(6) NULL,
27    PRIMARY KEY (`id_adres`))
28  ENGINE = InnoDB;
```

Rysunek 25: Wygenerowany fragment skryptu przez program - tworzenie tabeli adres

```
215  -----
216  -- Table `mydb`.`kurier_zna_adresy`
217  -----
218  CREATE TABLE IF NOT EXISTS `mydb`.`kurier_zna_adresy` (
219    `adres_id_adres` INT NOT NULL,
220    `kurier_id_kurier` INT NOT NULL,
221    PRIMARY KEY (`adres_id_adres`, `kurier_id_kurier`),
222    INDEX `fk_adres_has_kurier_kurier1_idx` (`kurier_id_kurier` ASC) VISIBLE,
223    INDEX `fk_adres_has_kurier_adres1_idx` (`adres_id_adres` ASC) VISIBLE,
224    CONSTRAINT `fk_adres_has_kurier_adres1`
225      FOREIGN KEY (`adres_id_adres`)
226      REFERENCES `mydb`.`adres` (`id_adres`)
227      ON DELETE NO ACTION
228      ON UPDATE NO ACTION,
229    CONSTRAINT `fk_adres_has_kurier_kurier1`
230      FOREIGN KEY (`kurier_id_kurier`)
231      REFERENCES `mydb`.`kurier` (`id_kurier`)
232      ON DELETE NO ACTION
233      ON UPDATE NO ACTION)
234  ENGINE = InnoDB;
```

Rysunek 26: Wygenerowany fragment skryptu przez program - tworzenie tabeli pośredniej

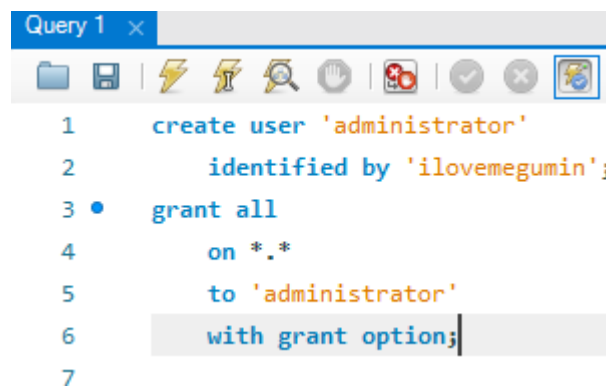
```

300  -----
301  -- View `mydb`.`klient`
302  -----
303  DROP TABLE IF EXISTS `mydb`.`klient`;
304  USE `mydb`;
305  CREATE OR REPLACE VIEW `klient` AS SELECT *
306  FROM `dane_osobowe` CROSS JOIN `uzytkownik` `adres`;
307
308  -----

```

Rysunek 27: Wygenerowany fragment skryptu przez program - tworzenie widoku klienta

Nadawanie uprawnień:

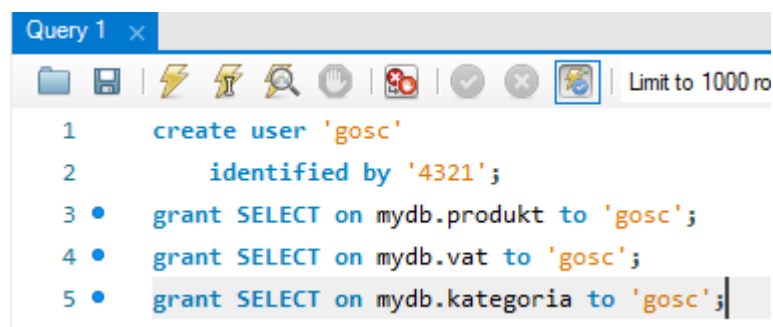


```

Query 1 x
1  create user 'administrator'
2      identified by 'ilovemegumin';
3  •  grant all
4      on *.*
5      to 'administrator'
6      with grant option;
7

```

Rysunek 28: Utworzenie konta administratora i nadanie mu uprawnień




```

Query 1 x
1  create user 'gosc'
2      identified by '4321';
3  •  grant SELECT on mydb.produkt to 'gosc';
4  •  grant SELECT on mydb.vat to 'gosc';
5  •  grant SELECT on mydb.kategoria to 'gosc';
Limit to 1000 ro


```

Rysunek 29: Utworzenia konta gościa i nadanie mu uprawnień



```
Query 1 x
1 • create user 'klient'
2   identified by '1234';
3 • grant SELECT on mydb.produkt to 'klient';
4 • grant SELECT on mydb.vat to 'klient';
5 • grant SELECT on mydb.faktura to 'klient';
6 • grant SELECT on mydb.kategoria to 'klient';
7 • grant SELECT, CREATE, UPDATE, DELETE on mydb.dane_osobowe to 'klient';
8 • grant SELECT, CREATE, UPDATE, DELETE on mydb.adres to 'klient';
9 • grant SELECT, UPDATE on mydb.uzytkownik to 'klient';
10 • grant SELECT, CREATE, UPDATE on mydb.transakcja to 'klient';
11 • grant SELECT, UPDATE, DELETE on mydb.koszyk to 'klient';
12 • grant SELECT, CREATE, UPDATE on mydb.platnosc to 'klient';
```

Rysunek 30: Utworzenie konta klient i nadanie mu uprawnień



```
Query 1 x
1 • create user 'serwis_transakcyjny'
2   identified by 'qwerty';
3 • grant SELECT, UPDATE on mydb.platnosc to 'serwis_transakcyjny';
```

Rysunek 31: Utworzenie konta serwis transakcyjny i nadanie mu uprawnień

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Wygląda to następująco:

✓	34	20:34:23	grant SELECT, CREATE, UPDATE, DELETE on mydb.dane_osobowe to 'klient'
✓	35	20:34:23	grant SELECT, CREATE, UPDATE, DELETE on mydb.adres to 'klient'
✓	36	20:34:23	grant SELECT, UPDATE on mydb.uzytkownik to 'klient'
✓	37	20:34:23	grant SELECT, CREATE, UPDATE on mydb.transakcja to 'klient'
✓	38	20:34:23	grant SELECT, UPDATE, DELETE on mydb.koszyk to 'klient'
✓	39	20:34:23	grant SELECT, CREATE, UPDATE on mydb.platnosc to 'klient'

Rysunek 32: Weryfikacja poprawności skryptu

Dodatkowo można sprawdzić nadane uprawnienia korzystając z terminala. Aby tego dokonać wpisujemy:

show grants for 'nazwa_naszego_uzytkownika'

i naszym oczom powinna się ukazać następująca tabelka (tutaj dla przykładu użyłem klienta).


```
mysql> show grants for klient
-> ;
+-----+
| Grants for klient@% |
+-----+
| GRANT USAGE ON *.* TO `klient`@`%` |
| GRANT SELECT, UPDATE, DELETE, CREATE ON `mydb`.`adres` TO `klient`@`%` |
| GRANT SELECT, UPDATE, DELETE, CREATE ON `mydb`.`dane_osobowe` TO `klient`@`%` |
| GRANT SELECT ON `mydb`.`faktura` TO `klient`@`%` |
| GRANT SELECT ON `mydb`.`kategoria` TO `klient`@`%` |
| GRANT SELECT, UPDATE, DELETE ON `mydb`.`koszyk` TO `klient`@`%` |
| GRANT SELECT ON `mydb`.`produkt` TO `klient`@`%` |
| GRANT SELECT, UPDATE, CREATE ON `mydb`.`płatność` TO `klient`@`%` |
| GRANT SELECT, UPDATE, CREATE ON `mydb`.`transakcja` TO `klient`@`%` |
| GRANT SELECT, UPDATE ON `mydb`.`uzytkownik` TO `klient`@`%` |
| GRANT SELECT ON `mydb`.`vat` TO `klient`@`%` |
+-----+
11 rows in set (0.00 sec)
```

Rysunek 33: Weryfikacja nadanych uprawnień przy wykorzystaniu terminala

4.2. Testy uprawnień

Administrator:

Aby wykonać test uprawnień należy zalogować się jak dany użytkownik do bazy danych:

Connect to Database

Stored Connection: **administrator** Select from saved connection settings

Connection Method: **Standard (TCP/IP)** Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: **127.0.0.1** Port: **3306** Name or IP address of the server host - and TCP/IP port.

Username: **administrator** Name of the user to connect with.

Password: **Store in Vault ...** **Clear** The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

OK **Cancel**

Rysunek 34: Logowanie się jako administrator

```

1 • INSERT INTO mydb.vat (id_vat, stawka_vat) VALUES('1', '23');
2 • SELECT * FROM mydb.vat;
3 • UPDATE mydb.vat SET stawka_vat='18' WHERE id_vat='1';
4 • SELECT * FROM mydb.vat;
5 • DELETE FROM mydb.vat WHERE id_vat='1';
6 • SELECT * FROM mydb.vat;

```

Rysunek 35: Sprawdzenie uprawnień Administratora dla tabeli „vat”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Wygląda to następująco:

#	Time	Action
✓ 1	11:56:10	INSERT INTO mydb.vat (id_vat, stawka_vat) VALUES('1', '23')
✓ 2	11:56:10	SELECT * FROM mydb.vat LIMIT 0, 1000
✓ 3	11:56:11	UPDATE mydb.vat SET stawka_vat='18' WHERE id_vat='1'
✓ 4	11:56:11	SELECT * FROM mydb.vat LIMIT 0, 1000
✓ 5	11:56:11	DELETE FROM mydb.vat WHERE id_vat='1'
✓ 6	11:56:11	SELECT * FROM mydb.vat LIMIT 0, 1000

Rysunek 36: Weryfikacja poprawności skryptu dla testu na TAK

	id_vat	stawka_vat
▶	1	23
*	NULL	NULL

Rysunek 37: Stan tabeli po dodaniu nowej stawki vat

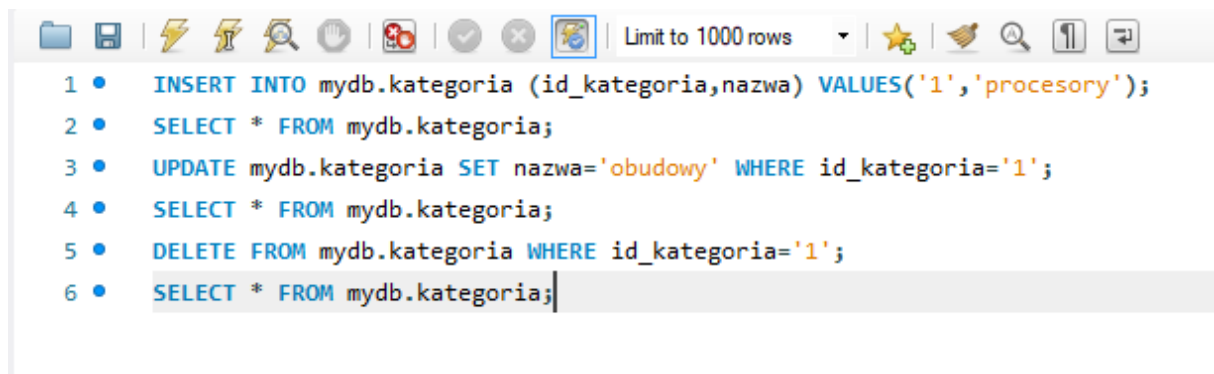
	id_vat	stawka_vat
▶	1	18
*	NULL	NULL

Rysunek 38: Stan tabeli po edycji stawki vat z indeksem 1

	id_vat	stawka_vat
*	NULL	NULL

Rysunek 39: Stan tabeli po usunięciu stawki vat z indeksem 1

Sprawdzenie uprawnień Administratora dla tabeli „kategoria”.

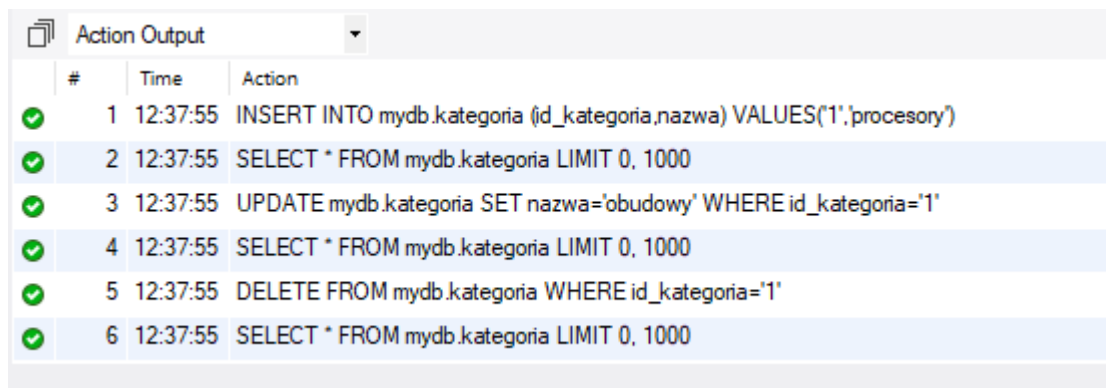


```

1 • INSERT INTO mydb.kategoria (id_kategoria,nazwa) VALUES('1','procesory');
2 • SELECT * FROM mydb.kategoria;
3 • UPDATE mydb.kategoria SET nazwa='obudowy' WHERE id_kategoria='1';
4 • SELECT * FROM mydb.kategoria;
5 • DELETE FROM mydb.kategoria WHERE id_kategoria='1';
6 • SELECT * FROM mydb.kategoria;
  
```

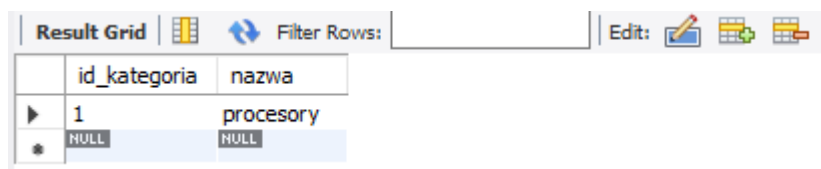
Rysunek 40: Przeprowadzanie testu uprawnień dla administratora dla tabeli "kategoria"

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Wygląda to następująco:



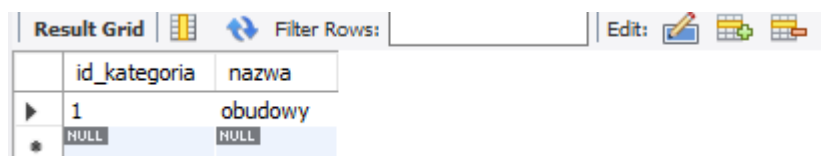
#	Time	Action
✓ 1	12:37:55	INSERT INTO mydb.kategoria (id_kategoria,nazwa) VALUES('1','procesory')
✓ 2	12:37:55	SELECT * FROM mydb.kategoria LIMIT 0, 1000
✓ 3	12:37:55	UPDATE mydb.kategoria SET nazwa='obudowy' WHERE id_kategoria='1'
✓ 4	12:37:55	SELECT * FROM mydb.kategoria LIMIT 0, 1000
✓ 5	12:37:55	DELETE FROM mydb.kategoria WHERE id_kategoria='1'
✓ 6	12:37:55	SELECT * FROM mydb.kategoria LIMIT 0, 1000

Rysunek 41: Weryfikacja poprawności skryptu dla testu na TAK



	id_kategoria	nazwa
▶	1	procesory
*	NULL	NULL

Rysunek 42: Stan tabeli po dodaniu nowej kategorii



	id_kategoria	nazwa
▶	1	obudowy
*	NULL	NULL

Rysunek 43: Stan tabeli po edycji kategorii z indeksem 1



	id_kategoria	nazwa
*	NULL	NULL

Rysunek 44: Stan tabeli po usunięciu kategorii z indeksem 1

Ostateczne wyniki testów uprawnień dla administratora:

	SELECT	CREATE	UPDATE	DELETE
adres	Tak	Tak	Tak	Tak
Dane_osobowe	Tak	Tak	Tak	Tak
egzemplarze	Tak	Tak	Tak	Tak
Faktura	Tak	Tak	Tak	Tak
kategoria	Tak	Tak	Tak	Tak
koszyk	Tak	Tak	Tak	Tak
kurier	Tak	Tak	Tak	Tak
płatość	Tak	Tak	Tak	Tak
produkt	Tak	Tak	Tak	Tak
transakcja	Tak	Tak	Tak	Tak
uzytkownik	Tak	Tak	Tak	Tak
vat	Tak	Tak	Tak	Tak

2 Wyniki testów uprawnień dla administratora

Gość:

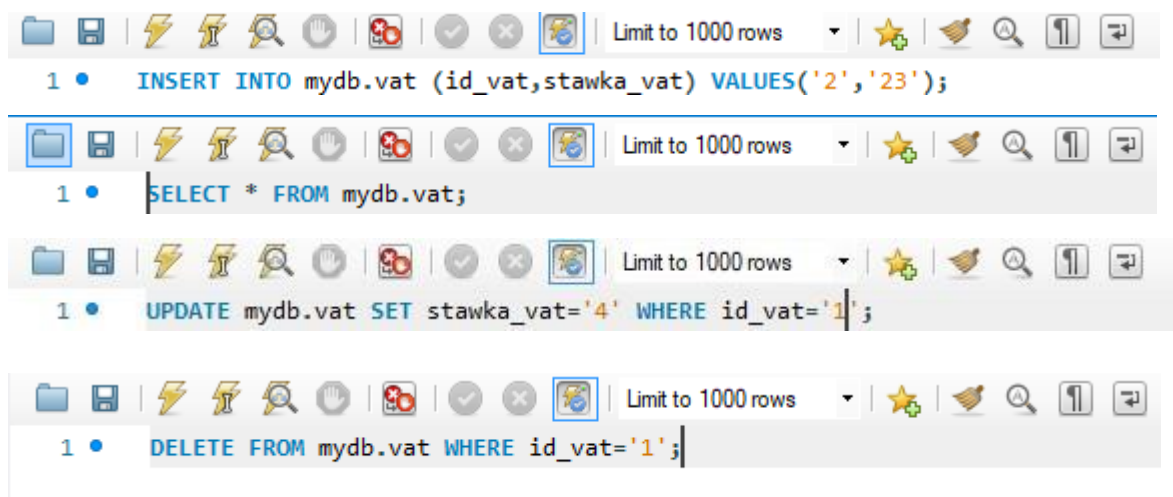
Aby wykonać test uprawnień należy zalogować się jak dany użytkownik do bazy danych:

The screenshot shows the 'Connect to Database' dialog box. The 'Stored Connection' dropdown is set to 'gosc'. The 'Connection Method' is 'Standard (TCP/IP)'. The 'Parameters' tab is active, showing the following fields:

- Hostname: 127.0.0.1
- Port: 3306
- Username: gosc
- Password: Store in Vault ... (button), Clear (button)
- Default Schema: (empty text box)

The 'OK' button is highlighted in blue.

Rysunek 45: Logowanie się jako gość



Rysunek 46: Sprawdzenie uprawnień Gościa dla tabeli „vat”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Jak widać Gość może jedynie odczytać dane z tabeli „vat”:

#	Time	Action	Message
✗	1 13:19:29	INSERT INTO mydb.vat (id_vat, stawka_vat) VALUES('1','23')	Error Code: 1142. INSERT command denied to user 'gosc'@'localhost' for table 'vat'
✓	2 13:19:34	SELECT * FROM mydb.vat LIMIT 0, 1000	1 row(s) returned
✗	3 13:19:37	UPDATE mydb.vat SET stawka_vat='4' WHERE id_vat='1'	Error Code: 1142. UPDATE command denied to user 'gosc'@'localhost' for table 'vat'
✗	4 13:19:42	DELETE FROM mydb.vat WHERE id_vat='1'	Error Code: 1142. DELETE command denied to user 'gosc'@'localhost' for table 'vat'

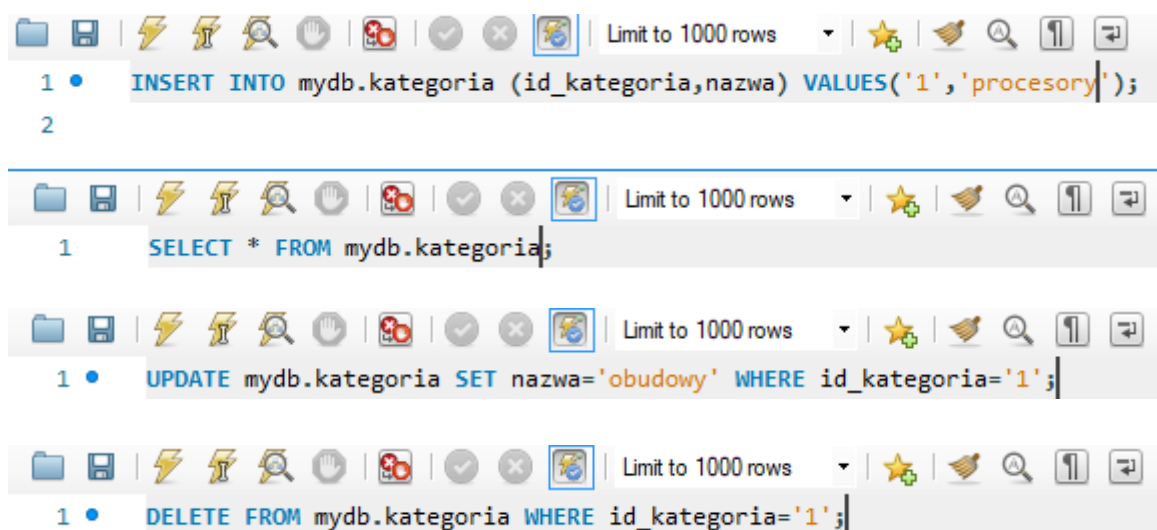
Rysunek 47: Ukazanie weryfikacji testu na TAK i na NIE

Result Grid			Filter Rows:	Edit:
	id_vat	stawka_vat		
▶	1	23		
*	NULL	NULL		

Rysunek 48: Stan tabeli „vat” przed

Result Grid			Filter Rows:	Edit:
	id_vat	stawka_vat		
▶	1	23		
*	NULL	NULL		

Rysunek 49: Stan tabeli „vat” po



Rysunek 50: Sprawdzenie uprawnień Gościa dla tabeli „kategoria”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Jak widać Gość może jedynie odczytać dane z tabeli „kategoria”:

Action Output				
#	Time	Action	Message	
✗	1 13:22:18	INSERT INTO mydb.kategoria (id_kategoria,nazwa) VALUES('1','procesory')	Error Code: 1142. INSERT command denied to user 'gosc'@'localhost' for table 'kategoria'	
✓	2 13:22:21	SELECT * FROM mydb.kategoria LIMIT 0, 1000	1 row(s) returned	
✗	3 13:22:24	UPDATE mydb.kategoria SET nazwa='obudowy' WHERE id_kategoria='1'	Error Code: 1142. UPDATE command denied to user 'gosc'@'localhost' for table 'kategoria'	
✗	4 13:22:26	DELETE FROM mydb.kategoria WHERE id_kategoria='1'	Error Code: 1142. DELETE command denied to user 'gosc'@'localhost' for table 'kategoria'	

Rysunek 51: Weryfikacja testu na TAK i na NIE

Result Grid			Filter Rows:	Edit:
	id_kategoria	nazwa		
▶	1	procesory		
*	NULL	NULL		

Rysunek 52: Stan tabeli „kategoria” przed

Result Grid			Filter Rows:	Edit:
	id_kategoria	nazwa		
▶	1	procesory		
*	NULL	NULL		

Rysunek 53: Stan tabeli „kategoria” po

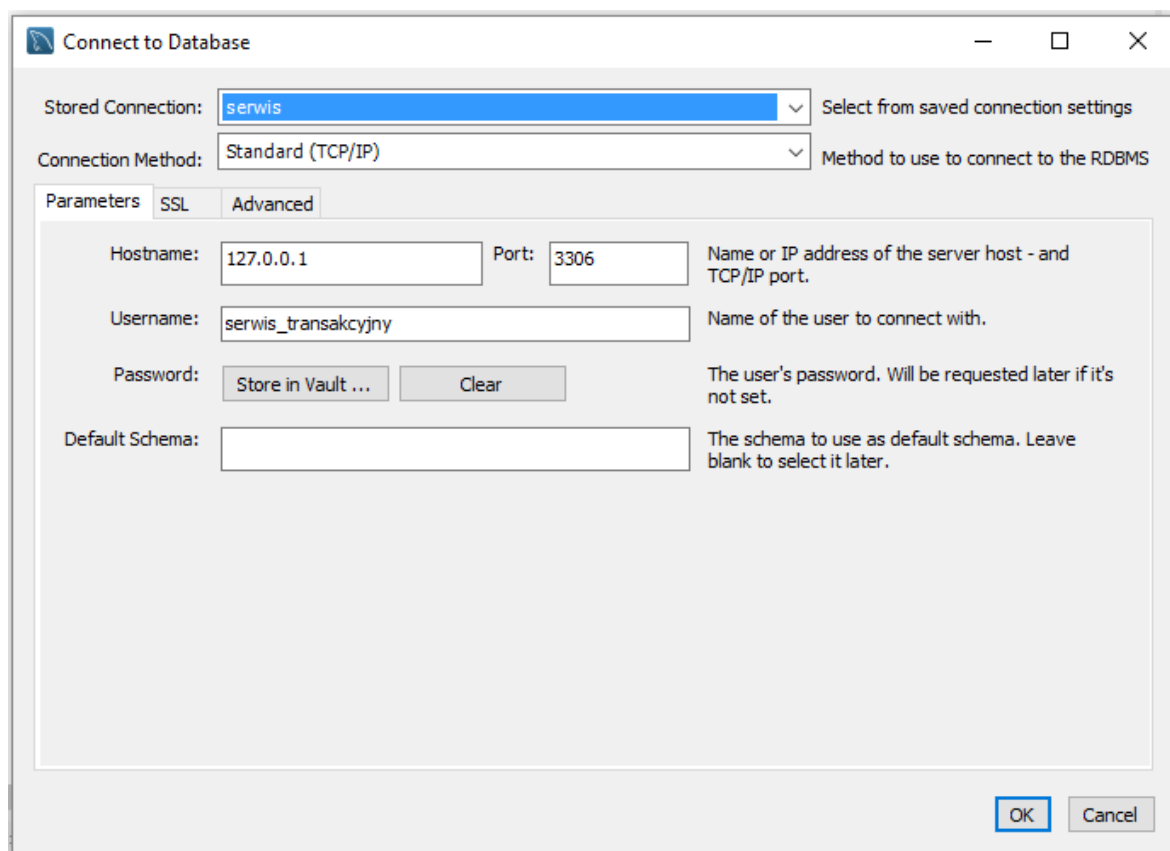
Ostateczne wyniki testów uprawnień dla gościa:

	SELECT	CREATE	UPDATE	DELETE
adres	Nie	Nie	Nie	Nie
Dane_osobowe	Nie	Nie	Nie	Nie
egzemplarze	Nie	Nie	Nie	Nie
Faktura	Nie	Nie	Nie	Nie
kategoria	Tak	Nie	Nie	Nie
koszyk	Nie	Nie	Nie	Nie
kurier	Nie	Nie	Nie	Nie
płatość	Nie	Nie	Nie	Nie
produkt	Tak	Nie	Nie	Nie
transakcja	Nie	Nie	Nie	Nie
uzytkownik	Nie	Nie	Nie	Nie
vat	Tak	Nie	Nie	Nie

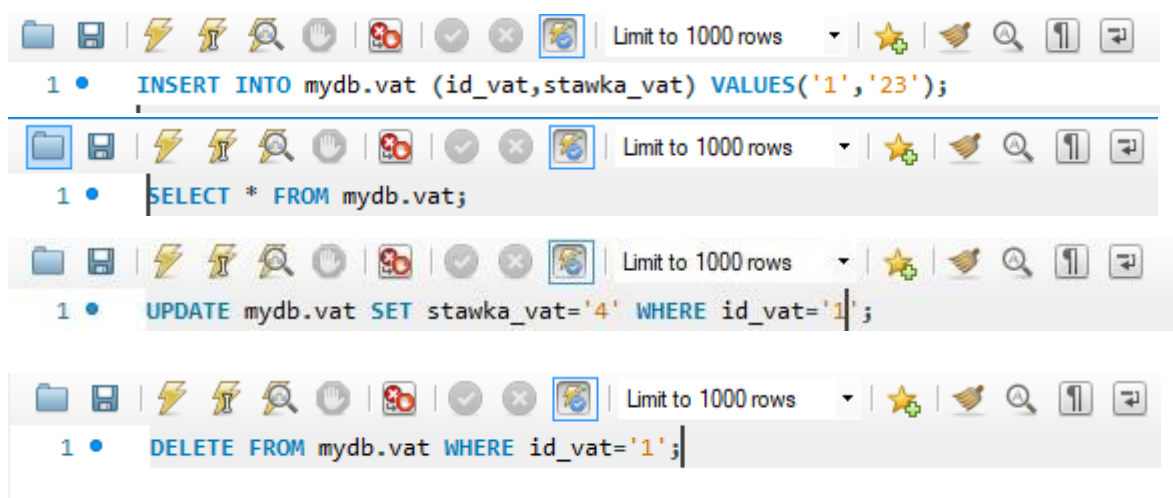
3 Wyniki testów uprawnień dla gościa

Serwis transakcyjny:

Aby wykonać test uprawnień należy zalogować się jak dany użytkownik do bazy danych:



Rysunek 54: Logowanie się jako serwis transakcyjny



Rysunek 55: Sprawdzenie uprawnień Serwisu transakcyjnego dla tabeli „vat”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Jak widać Serwis transakcyjny nie ma żadnych uprawnień do tabeli „vat”:

Action Output				
#	Time	Action	Message	
1	13:48:18	INSERT INTO mydb.vat (id_vat, stawka_vat) VALUES('1', '23')	Error Code: 1142. INSERT command denied to user 'serwis_transakcyjny'@'localhost' for table 'vat'	
2	13:48:19	SELECT * FROM mydb.vat LIMIT 0, 1000	Error Code: 1142. SELECT command denied to user 'serwis_transakcyjny'@'localhost' for table 'vat'	
3	13:48:21	UPDATE mydb.vat SET stawka_vat='4' WHERE id_vat='1'	Error Code: 1142. UPDATE command denied to user 'serwis_transakcyjny'@'localhost' for table 'vat'	
4	13:48:23	DELETE FROM mydb.vat WHERE id_vat='1'	Error Code: 1142. DELETE command denied to user 'serwis_transakcyjny'@'localhost' for table 'vat'	

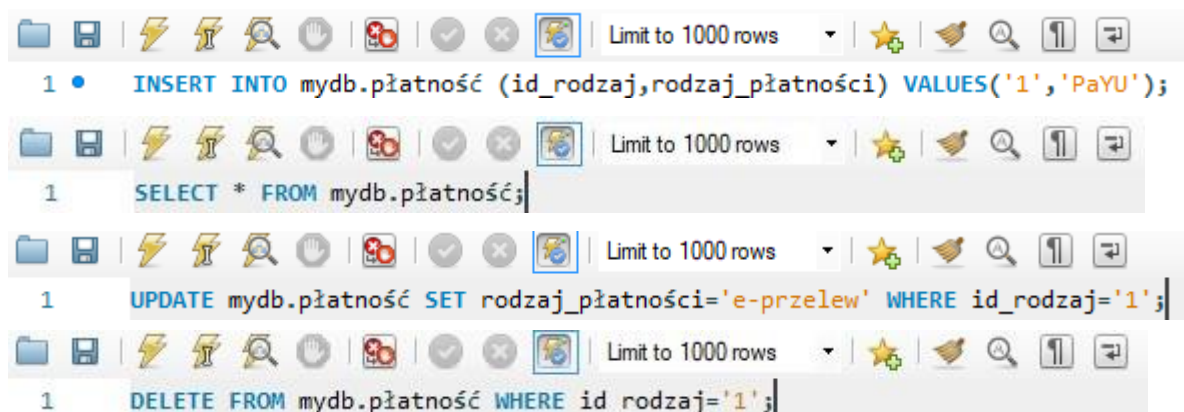
Rysunek 56: Weryfikacja testu na TAK

Result Grid			Filter Rows:	Edit:
	id_vat	stawka_vat		
▶	1	23		
*	NULL	NULL		

Rysunek 57: Stan tabeli „vat” przed

Result Grid			Filter Rows:	Edit:
	id_vat	stawka_vat		
▶	1	23		
*	NULL	NULL		

Rysunek 58: Stan tabeli „vat” po



Rysunek 59: Sprawdzenie uprawnień Serwisu transakcyjnego dla tabeli „płatność”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Jak widać Serwis transakcyjny może odczytać oraz edytować dane z tabeli „płatność”:

#	Time	Action	Message
✗ 1	13:59:06	INSERT INTO mydb.płatność (id_rodzaj, rodzaj_płatności) VALUES('1', 'PaYU')	Error Code: 1142. INSERT command denied to user 'serwis_transakcyjny'@'localhost' for table 'płatno
✓ 2	13:59:09	SELECT * FROM mydb.płatność LIMIT 0, 1000	1 row(s) returned
✓ 3	13:59:26	UPDATE mydb.płatność SET rodzaj_płatności='e-przelew' WHERE id_rodzaj='1'	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
✗ 4	13:59:39	DELETE FROM mydb.płatność WHERE id_rodzaj='1'	Error Code: 1142. DELETE command denied to user 'serwis_transakcyjny'@'localhost' for table 'płatno

Rysunek 60: Weryfikacja testu na TAK i na NIE

id_rodzaj	rodzaj_płatności	adres_strony_platniczej	kod_platnosci
1	PaYU	NULL	NULL
NULL	NULL	NULL	NULL

Rysunek 61: Stan tabeli „płatność” przed

id_rodzaj	rodzaj_płatności	adres_strony_platniczej	kod_platnosci
1	e-przelew	NULL	NULL
NULL	NULL	NULL	NULL

Rysunek 62: Stan tabeli „płatność” po

Ostateczne wyniki testów uprawnień dla serwisu transakcyjnego:

	SELECT	CREATE	UPDATE	DELETE
adres	Nie	Nie	Nie	Nie
Dane_osobowe	Nie	Nie	Nie	Nie
egzemplarze	Nie	Nie	Nie	Nie
Faktura	Nie	Nie	Nie	Nie
kategoria	Nie	Nie	Nie	Nie
koszyk	Nie	Nie	Nie	Nie
kurier	Nie	Nie	Nie	Nie
płatość	Tak	Nie	Tak	Nie
produkt	Nie	Nie	Nie	Nie
transakcja	Nie	Nie	Nie	Nie
uzytkownik	Nie	Nie	Nie	Nie
vat	Nie	Nie	Nie	Nie

4 Wyniki testów uprawnień dla serwisu transakcyjnego

Klient:

Aby wykonać test uprawnień należy zalogować się jak dany użytkownik do bazy danych:

Connect to Database

Stored Connection: klient Select from saved connection settings

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host - and TCP/IP port.

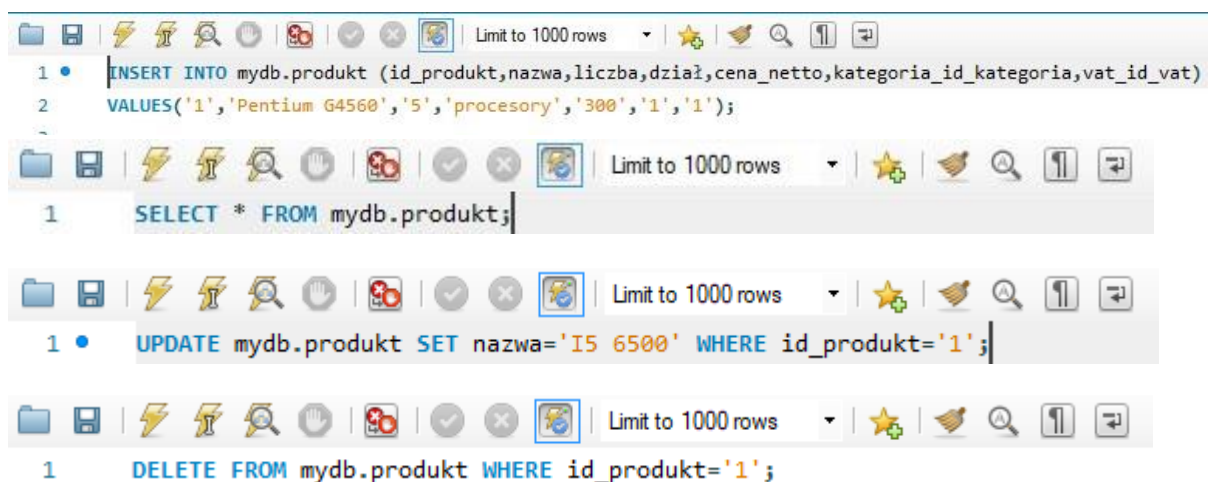
Username: klient Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

OK Cancel

Rysunek 63: Logowanie się jako klient



Rysunek 64: Sprawdzenie uprawnień Serwisu transakcyjnego dla tabeli „produkt”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Jak widać Klient może jedynie odczytywać dane z tabeli „produkt”:

#	Time	Action	Message
✗	1 14:26:17	INSERT INTO mydb.produkt (id_produk,nazwa,liczba,dzia,cena_netto,kategoria_id_kategoria,vat_id_vat)...	Error Code: 1142. INSERT command denied to user 'klient'@'localhost' for table 'produkt'
✓	2 14:26:22	SELECT * FROM mydb.produkt LIMIT 0, 1000	1 row(s) returned
✗	3 14:26:31	UPDATE mydb.produkt SET nazwa='I5 6500' WHERE id_produk='1'	Error Code: 1142. UPDATE command denied to user 'klient'@'localhost' for table 'produkt'
✗	4 14:26:35	DELETE FROM mydb.produkt WHERE id_produk='1'	Error Code: 1142. DELETE command denied to user 'klient'@'localhost' for table 'produkt'

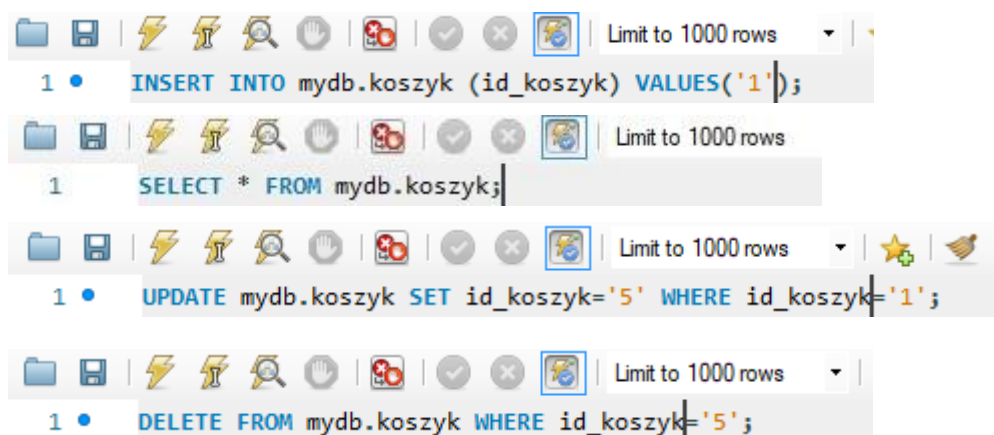
Rysunek 65: Weryfikacja testu na TAK i na NIE

	id_produk	nazwa	liczba	dzia	cena_netto	kategoria_id_kategoria	vat_id_vat
▶	1	Pentium G4560	5	procesory	300	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 66: Stan tabeli „produkt” przed

	id_produk	nazwa	liczba	dzia	cena_netto	kategoria_id_kategoria	vat_id_vat
▶	1	Pentium G4560	5	procesory	300	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 67: Stan tabeli „produkt” po



Rysunek 68: Sprawdzenie uprawnień Klienta dla tabeli „koszyk”

Program MySQL Workbench informuje, czy skrypt został wykonany poprawnie poprzez zielony znaczek weryfikacji przy każdej komendzie. Jak widać Klient może odczytać, edytować oraz usunąć dane z tabeli „koszyk”:

Action Output				
#	Time	Action	Message	
✖	1 15:02:32	INSERT INTO mydb.koszyk (id_koszyk) VALUES('1')	Error Code: 1142. INSERT command denied to user 'Klient'@'localhost' for table 'koszyk'	
✔	2 15:02:38	SELECT * FROM mydb.koszyk LIMIT 0, 1000	1 row(s) returned	
✔	3 15:02:41	UPDATE mydb.koszyk SET id_koszyk='5' WHERE id_koszyk='1'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	
✔	4 15:02:44	DELETE FROM mydb.koszyk WHERE id_koszyk='5'	1 row(s) affected	

Rysunek 69: Weryfikacja testu na TAK i na NIE

Result Grid		Filter Rows:	Edit:
	id_koszyk		
▶	1		
*	NULL		

Rysunek 70: Stan tabeli „koszyk” przed

Result Grid		Filter Rows:	Edit:
	id_koszyk		
▶	5		
*	NULL		

Rysunek 71: Stan tabeli „koszyk” po

Ostateczne wyniki testów uprawnień dla klienta:

	SELECT	CREATE	UPDATE	DELETE
adres	Tak	Tak	Tak	Tak
Dane_osobowe	Tak	Tak	Tak	Tak
egzemplarze	Nie	Nie	Nie	Nie
Faktura	Tak	Nie	Nie	Nie
kategoria	Tak	Nie	Nie	Nie
koszyk	Tak	Nie	Tak	Tak
kurier	Nie	Nie	Nie	Nie
płatność	Tak	Tak	Tak	Nie
produkt	Tak	Nie	Nie	Nie
transakcja	Tak	Tak	Tak	Nie
uzytkownik	Tak	Nie	Tak	Nie
vat	Tak	Nie	Nie	Nie

5

Wyniki testów uprawnień dla serwisu klienta

By przetestować dodatkowo wydajność naszego systemu dodaliśmy 1000 danych losowo generowanych poprzez stronę internetową www.mockaroo.com. Do tego by przetestować stabilność pól NULL. Wprowadziliśmy je w z 20% szansą w dane, które nie są obowiązkowe do wprowadzenia.

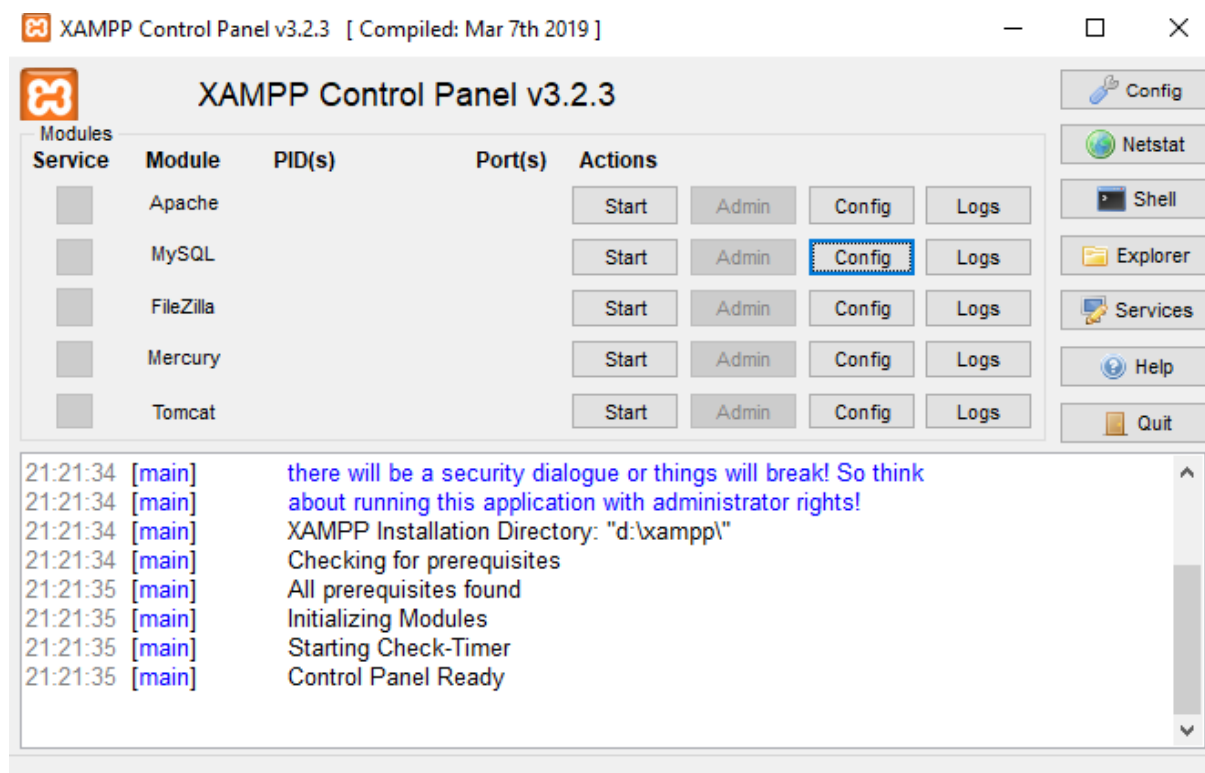
id_dane_osobowe	nazwa_firmy	nip	imię	nazwisko	nr_telefonu	email
1	NULL	NULL	Cobb	Barnbrook	NULL	cbarnbrook0@i2i.jp
2	NULL	NULL	Elwyn	Offer	5178576354	eoffer1@devhub.com
3	NULL	NULL	Georgetta	Adshed	6962138230	gadshed2@gov.uk
4	Bergnaum, Skiles and Roob	558	Lorette	Pride	1357013028	lpride3@vix.com
5	NULL	NULL	Angele	Hugenin	1872191508	ahugenin4@symantec.com
6	NULL	NULL	Ruthann	Pauletti	NULL	rpauletti5@timesonline.co.uk
7	NULL	NULL	Whitaker	Kroin	7327886133	wkroin6@quantcast.com
8	NULL	NULL	Hally	Coils	4951461035	hcoils7@gravatar.com
9	NULL	NULL	Claudian	Gerge	4992810697	cgerge8@ycombinator.com
10	Murphy Inc	312	Banky	Biggadyke	7781593155	bbiggadyke9@jugem.jp
11	NULL	NULL	Harold	Zelake	NULL	hzalakea@bbb.org
12	NULL	NULL	Cassius	Jery	5682156099	cjeryb@cloudflare.com
13	Boyle Inc	174	Maddi	Molohan	2878184567	mmolohanc@wp.com
14	Weissnat-Breitenberg	822	Alexandre	Groom	NULL	agroomd@imdb.com
15	NULL	NULL	Rogers	Jaksic	2697378738	rkajsice@blinklist.com
16	NULL	NULL	Bronnie	Devigne	8902013429	bdevignef@storify.com
17	NULL	NULL	Eulalie	O'Curine	3611504086	eoocurrineg@slashdot.org
18	NULL	NULL	Hilliary	Beltzner	3123425971	hbeltznerh@xrea.com
19	NULL	NULL	Eduardo	Doey	NULL	edoeyl@soup.io
20	NULL	NULL	Debby	Alenichev	NULL	dalenichevj@elegantthemes.com
21	Stracke, Wuckert and Veum	215	Mitchel	Bibb	NULL	mbibbk@wired.com
22	NULL	NULL	Caritta	Dace	9836349482	cdacel@examiner.com
23	NULL	NULL	Lisetta	Rougier	5573316267	lrougierm@washington.edu
24	NULL	NULL	Avictor	Avory	NULL	aavoryn@mac.com
25	NULL	NULL	Isador	Beric	NULL	iberico@youtube.com

Rysunek 72: Dane osobowe wprowadzone losowo

5. Implementacja i testy aplikacji

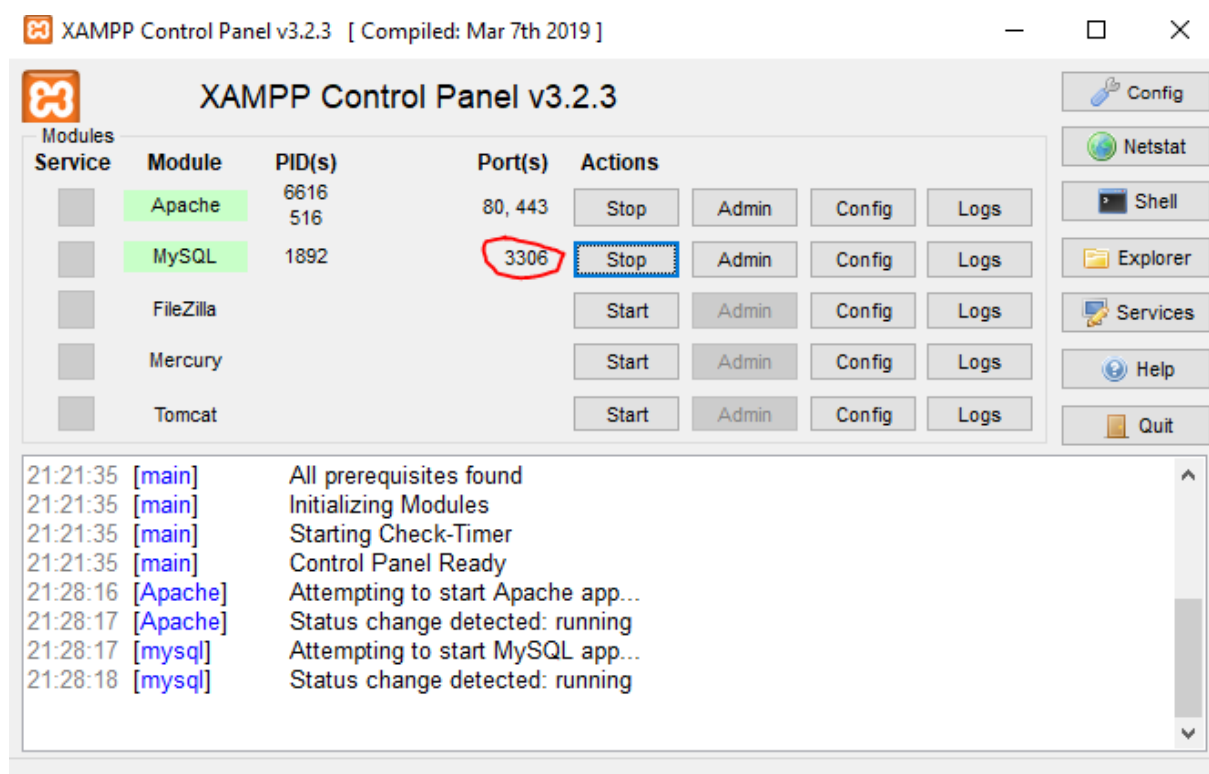
5.1. Instalacja i konfigurowanie systemu

Aby móc napisać aplikację dla naszej bazy danych na samym początku należało postawić serwer gdzie owa baza oraz kod aplikacji miały się znajdować. Tutaj postawiliśmy na czas testów tego etapu na serwer lokalny. Pakietem, który spełnił wszystkie nasze potrzeby był XAMPP, czyli całkowicie darmowa dystrybucja Apache'a, zawierająca MariaDB oraz PHP. Proces instalacji jest dość prosty a w razie jakiś większych problemów można udać się na ich stronę główną, gdzie cały proces jest opisany krok po kroku. Po zainstalowaniu pakietu naszym oczom ukaże się panel kontrolny:



Rysunek 73: Panel kontrolny pakietu XAMPP

Kolejnym krokiem było nawiązanie połączenia pomiędzy programem MySQL Workbench a XAMPP'em w celu wyeksportowania naszej bazy na serwer. Dokuje się tego poprzez naciśnięciu na panelu kontrolnym przycisku start na module MySQL. Najistotniejszą informacją jaką możemy odczytać po wciśnięciu tego przycisku jest Port, który będzie nam potrzebny do nawiązania wcześniej wspomnianego połączenia.



Rysunek 74: Widok panelu kontrolnego po uruchomieniu interesujących nas modułów

Następnie eksportujemy naszą bazę na serwer korzystając z funkcji Forward Engineer. Po wybraniu tej funkcji wyskoczy nam okno, gdzie należy pamiętać aby uzupełnić poprawnie parametry wykorzystując wcześniej wspomniany Port.

Set Parameters for Connecting to a DBMS

Stored Connection: Local instance MySQL80 Select from saved connection settings

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: localhost Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Rysunek 75: Ustawienie parametrów w MySQL Workbench

Reszta operacji została opisana krok po kroku w dokumentacji do etapu trzeciego, dlatego pozwolę sobie pominąć większą część procesu i przejdę od razu do meritum. Jednym z problemów, które można tutaj napotkać jest generacja przez Workbench'a indexów dla kluczy obcych oraz constraintów, które nie występują nigdzie w bazie. Jest to problem dość często spotykany, który generuje później nieścisłości w tworzonych tablicach przez co nie można ich wypełniać. Dlatego najlepszym rozwiązaniem jest po prostu usunięcie ich już w momencie generacji kodu. Przykładowe linie kodu wymagające usunięcia wyglądają następująco:

```
INDEX `fk_transkacja_klient1_idx` (`uzytkownik_id_uzytkownik` ASC) VISIBLE,  
INDEX `fk_transkacja_faktura1_idx` (`faktura_id_faktura` ASC) VISIBLE,  
INDEX `fk_transkacja_platnosc1_idx` (`platnosc_id_rodzaj` ASC) VISIBLE,  
CONSTRAINT `fk_transkacja_klient1`
```

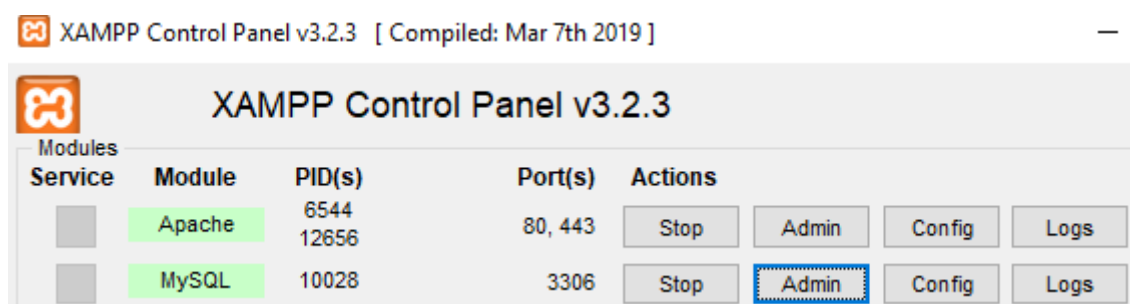
Rysunek 76: Błędna generacja indexów przez Workbench'a

O pomyślnym zakończeniu się całego procesu poinformuje nas następujący komunikat:



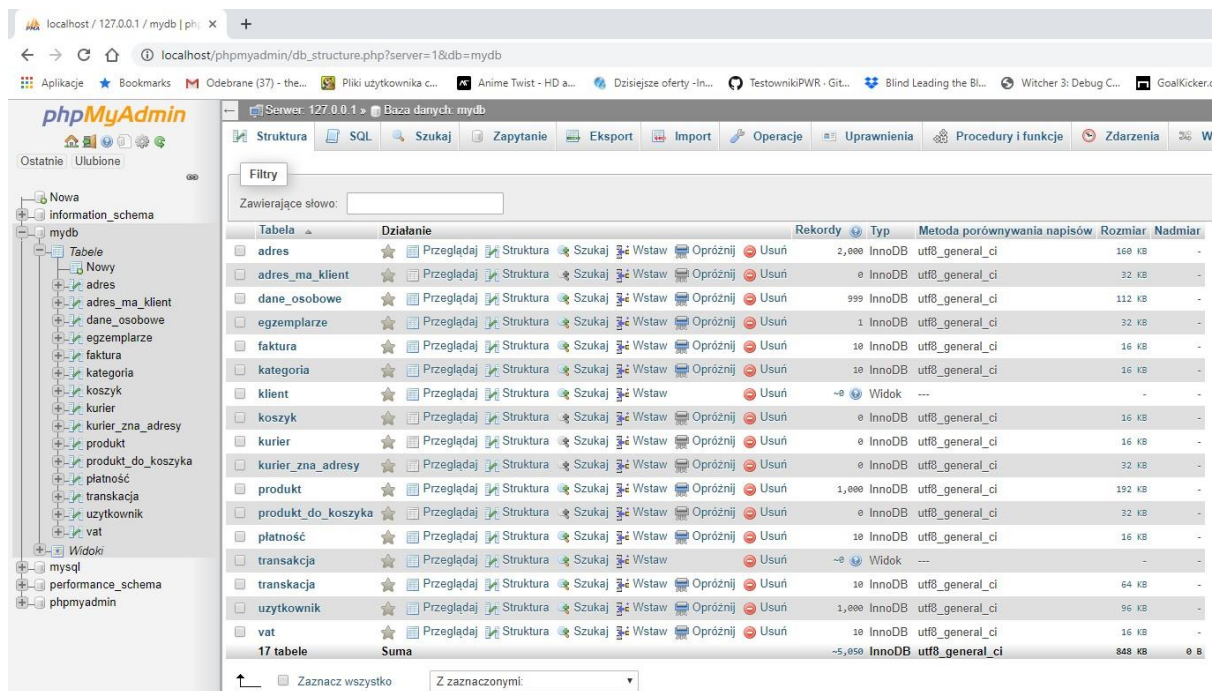
Rysunek 77: Pomyślne zakończenie się procesu Forward Engineer

W celu sprawdzenia uzyskanych rezultatów należy kliknąć przycisk Admin na panelu kontrolnym XAMPPa co przeniesie nas na stronę phpmyadmin, gdzie możemy zarządzać swoją bazą jako root.



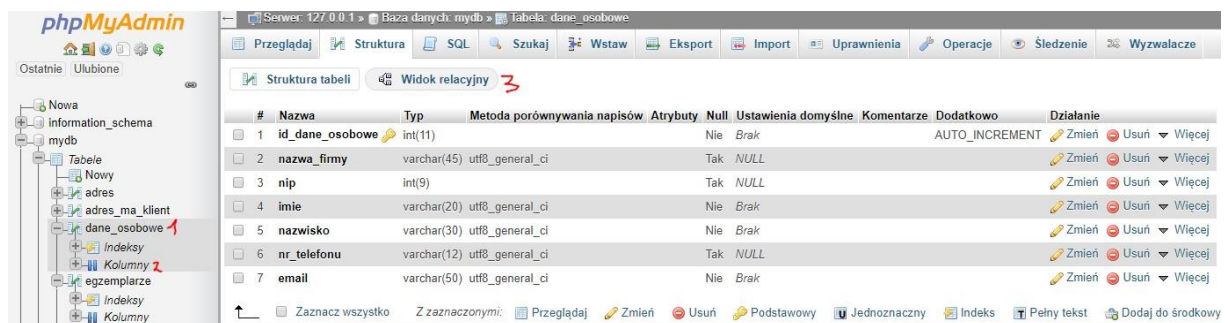
Rysunek 78: Panel kontrolny XAMPPa - przejście do zarządzania bazą

Naszym oczom powinien ukazać się następujący widok:



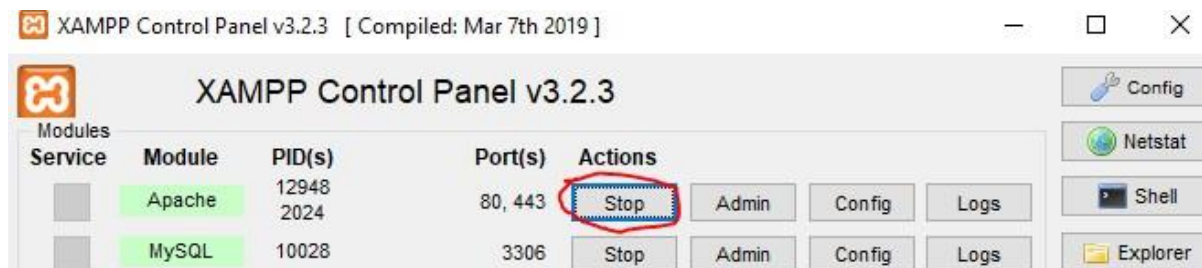
Rysunek 79: Widok bazy danych z poziomu phpmyadmin

Oczywiście na powyższym obrazku moja baza ma już wypełnione rekordy, nowo utworzona powinna być pusta. W celu jej zapełnienia można wykorzystać specjalnie przeznaczone do takich zadań strony jak: www.mockaroo.com, która została wykorzystana przez nas między innymi do przeprowadzenia testów wydajnościowych oraz na NOT NULLe. Przy egzekwowaniu wygenerowanych skryptów mogą wystąpić problemy powiązane z ograniczeniami kluczy obcych. Aby tego uniknąć należy wejść w interesującą nas tabelę, potem wybrać po lewej stronie opcję Kolumny oraz widok relacyjny. Następnie należy usunąć ograniczenie, które generuje błędy.



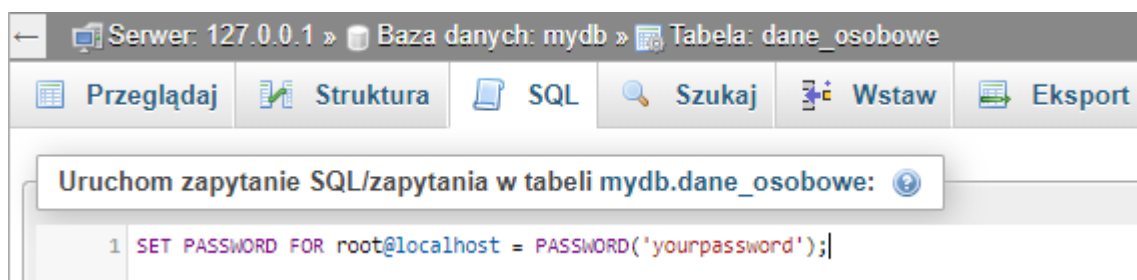
Rysunek 80: Usunięcie ograniczeń kluczy obcych

Ostatnimi krokami konfiguracyjnymi będzie uruchomienie serwera Apache, który uruchamia się przy pomocy naciśnięcia przycisku start obok jego nazwy oraz ustawienie hasła dla naszego serwera MySQL w celach bezpieczeństwa.



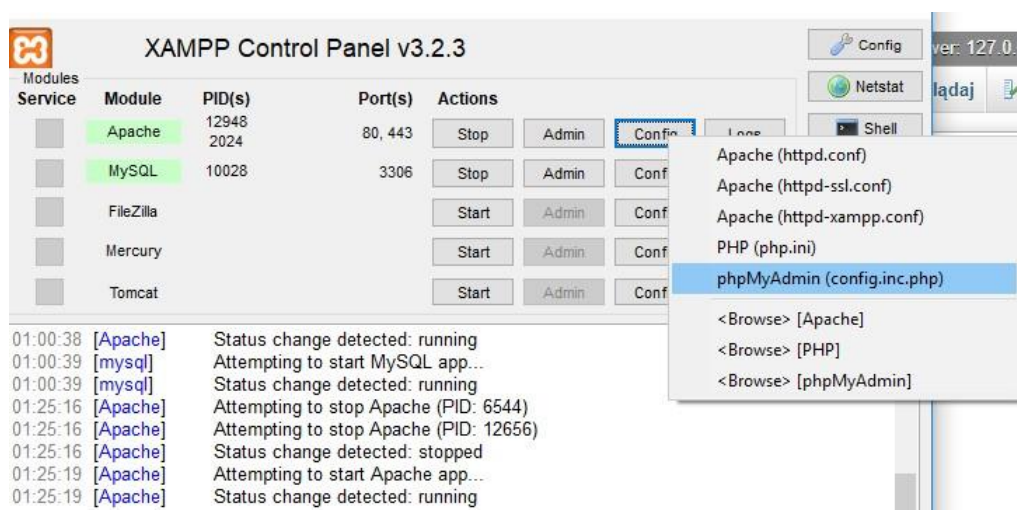
Rysunek 81: Uruchamianie serwera Apache

Aby ustawić hasło na samym początku należy wpisać zapytanie na stronie phpmyadmin w zakładce SQL:



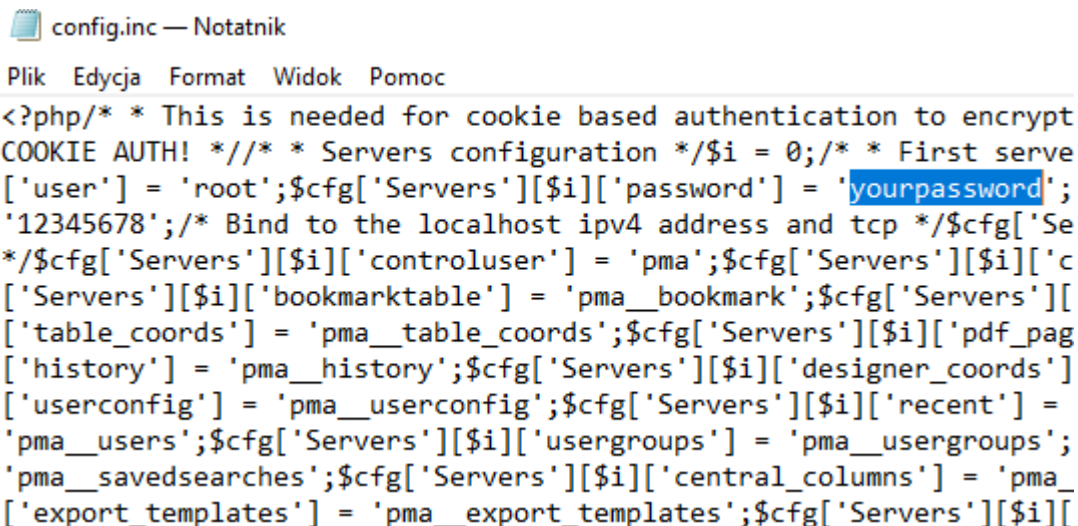
Rysunek 82: Ustawianie hasła dla bazy MySQL

Następnie klikamy wykonaj. Potem w ustawieniach serwera Apache wybieramy plik phpMyAdmin (config.inc.php).



Rysunek 83: Konfiguracja hasła z poziomu panelu kontrolnego

Po wybraniu tego pliku wyświetli nam się następujący plik konfiguracyjny. W zaznaczonym przeze mnie polu należy wpisać to samo hasło, które uprzednio ustawiliśmy na Rysunku 10.

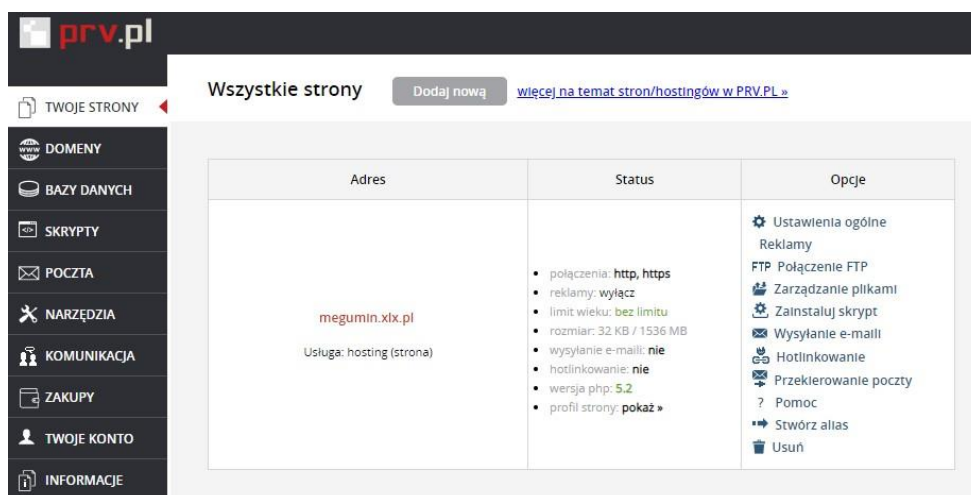


```
<?php/* * This is needed for cookie based authentication to encrypt
COOKIE AUTH! *//* * Servers configuration */$i = 0;/* * First serve
['user'] = 'root';$cfg['Servers'][$i]['password'] = 'yourpassword';
'12345678';/* Bind to the localhost ipv4 address and tcp */$cfg['Se
*/$cfg['Servers'][$i]['controluser'] = 'pma';$cfg['Servers'][$i]['c
['Servers'][$i]['bookmarktable'] = 'pma__bookmark';$cfg['Servers'][$
['table_coords'] = 'pma__table_coords';$cfg['Servers'][$i]['pdf_pag
['history'] = 'pma__history';$cfg['Servers'][$i]['designer_coords']
['userconfig'] = 'pma__userconfig';$cfg['Servers'][$i]['recent'] =
'pma__users';$cfg['Servers'][$i]['usergroups'] = 'pma__usergroups';
'pma__savedsearches';$cfg['Servers'][$i]['central_columns'] = 'pma_
['export_templates'] = 'pma__export_templates';$cfg['Servers'][$i][
```

Rysunek 84: Ustawienie hasła - plik konfiguracyjny

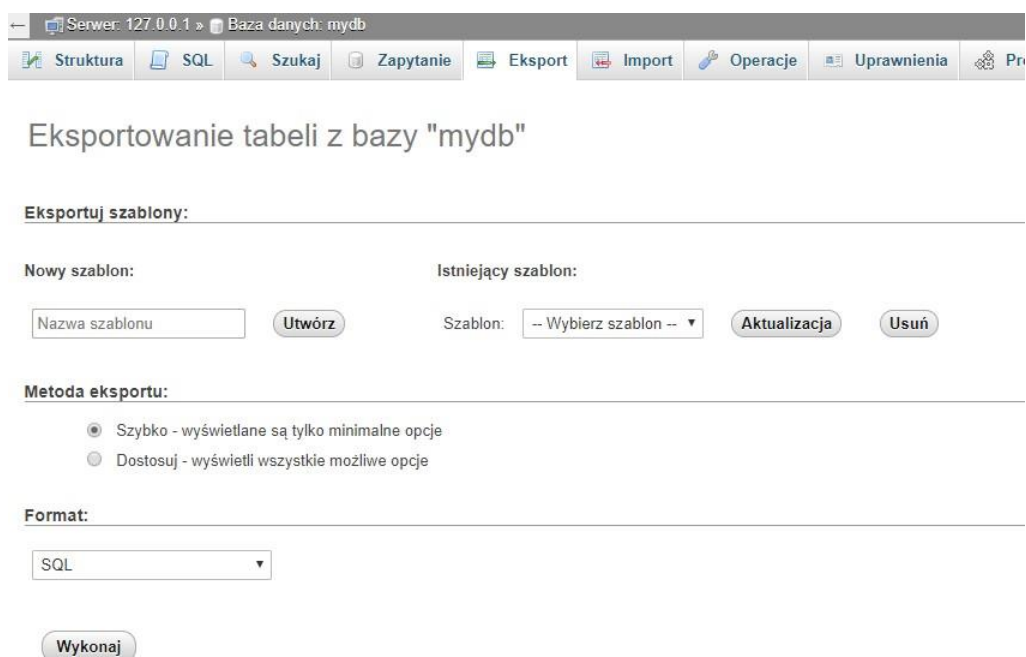
1. Konfiguracja hosta webowego:

Elementem dodatkowym podczas tego etapu było przeniesienie naszej aplikacji z serwera lokalnego na prawdziwy hosting webowy, który zapewni dostęp do naszej aplikacji z poziomu przeglądarek internetowych oraz urządzeń mobilnych wszystkim użytkownikom. Z racji tego, iż nie chcieliśmy ponosić żadnych kosztów postawiliśmy na darmowy hosting w zamian za możliwość wyświetlania reklam. Wybrana przez nas serwis to: www.prv.pl, na którym należało się zarejestrować podając adres e-mail, nazwę witryny, domenę oraz hasło, które posłuży nam za logowanie się do panelu administratora. Po zalogowaniu się naszym oczom ukaże się następujący widok:



Rysunek 85: Panel administratora na stronie hostingowej

W celu dodania nowej bazy danych należy najechać myszką na bazy danych, a następnie wybrać opcję dodaj nową bazę. Jakby dana opcja się nie pojawiała i cały pasek po lewej stronie nie był interaktywny należy wyłączyć Ad block'a i odświeżyć stronę. Po wybraniu tej opcji zostanie nam przydzielona nazwa bazy, nazwa użytkownika oraz hasło, które posłuży nam do zalogowania się na phpmyadmin. Po wykonaniu tej czynności możemy przejść do zaimportowania naszej bazy danych, lecz najpierw należy ją wyeksportować z naszego serwera lokalnego. W tym celu klikamy nazwę naszej bazy, potem dajemy zaznaczyć wszystko i przechodzimy do zakładki eksportuj. Należy pamiętać aby jako format pliku wybrać SQL.



Rysunek 86: Eksportowanie bazy danych z serwera lokalnego

Wybrany przez nas serwis hostingowy nie obsługuje nowszych wersji MySQL, dlatego warto mieć na uwadze, iż niektóre funkcje nie będą działać poprawnie na przykład widoki. Kolejnym krokiem jest import wygenerowanej bazy na nasz serwis. Dokonuje się tego poprzez utworzenia jakiejś przykładowej tabeli a następnie naciśnięcie przycisku Import znajdującego się w na górze strony. Naszym oczom ukaże się następujące okno:

Import do tabeli "1"

Plik do importu:

Plik może być skompresowany (gzip, bzip2, zip) bądź nie.
Plik skompresowany musi mieć rozszerzenie `.[format].[kompresja]`, np. `.sql.zip`

Wyszukaj w komputerze: Nie wybrano pliku (Maksymalny rozmiar: 8,192KB)

Kodowanie znaków pliku:

Import częściowy:

☒ Zezwól na przerwanie importu w przypadku, gdy skrypt wykryje zbliżający się koniec limitu czasu. (Może to być dobry sposób importu dużych plików, jednakże może on popsuć transakcje.)

Liczba wierszy, by pominąć, począwszy od pierwszego wiersza:

Format:

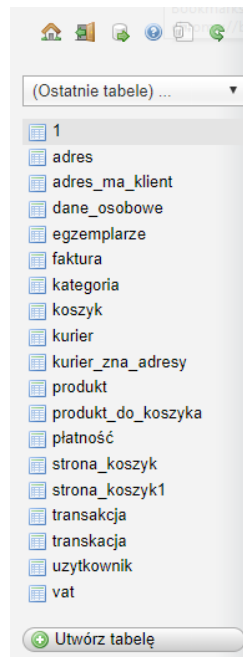
Opcje formatu:

Tryb zgodności SQL:

☒ Nie używaj `AUTO_INCREMENT` dla wartości zerowych

Rysunek 87: Import bazy danych na serwer hostingowy

Tutaj należy wybrać wygenerowany wcześniej skrypt, upewnić się czy naszym formatem jest SQL, a następnie klikamy przycisk wykonaj. O udanej operacji powiadomi nas komunikat w phpmyadmin, a po lewej stronie powinny pojawić się te same tablice, które uprzednio znajdowały się na serwerze lokalnym.



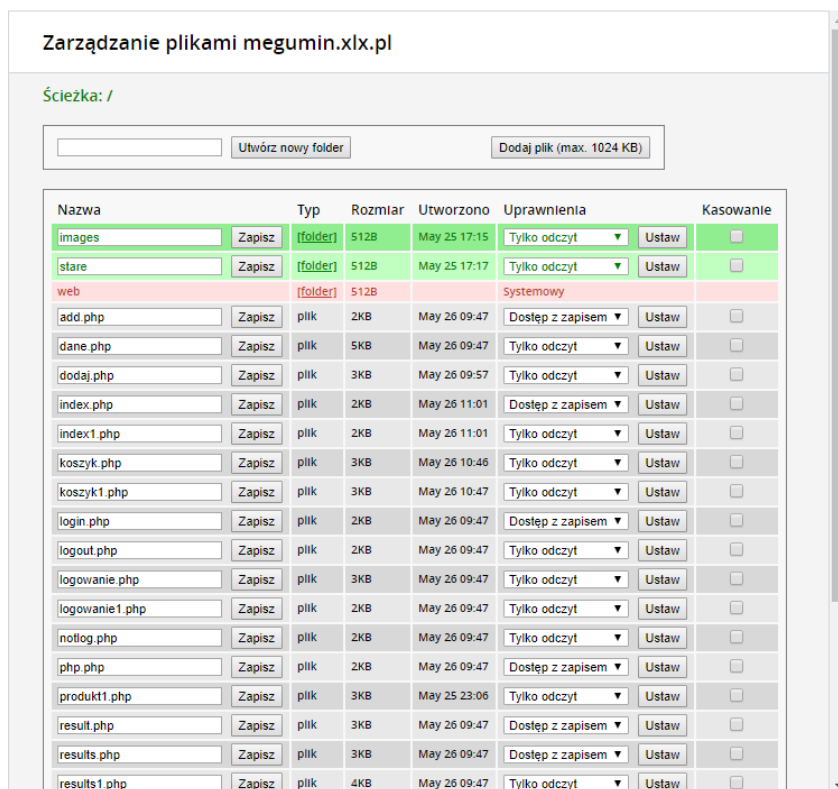
Rysunek 88: Widok zaimportowanych tabeli

Ostatnim krokiem jest umieszczenie skryptów napisanych w php, które odpowiadają za działanie naszej aplikacji. Dokonuje się tego poprzez wybranie opcje Zarządzanie plikami w zakładce twoje strony.

Adres	Status	Opcje
megumin.xlsx.pl	<ul style="list-style-type: none"> połączenia: http, https reklamy: wyłącz limit wieku: bez limitu rozmiar: 32 KB / 1536 MB 	<ul style="list-style-type: none"> Ustawienia ogólne Reklamy FTP Połączenie FTP <u>Zarządzanie plikami</u> Zainstaluj skrypt Wysyłanie e-maili

Rysunek 89: Zarządzanie skryptami

Następnie klikamy dodaj plik i zapisujemy odpowiednie pliki. Przykładowo będzie ten panel wyglądać następująco:

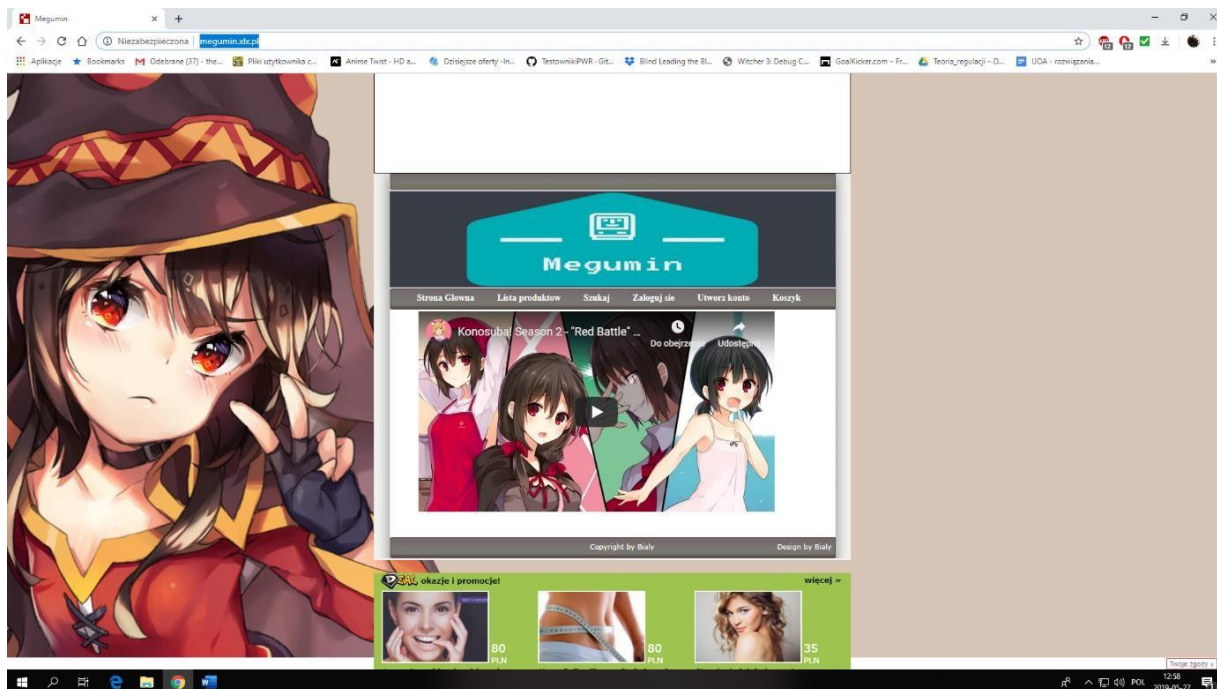


Rysunek 90: Dodawanie skryptów w języku php

Jeśli cała procedura przebiegła pomyślnie po wpisaniu wybranej przez siebie nazwy domeny dowolny użytkownik, który ma dostęp do Internetu powinien móc odwiedzić twoją aplikację webową.

5.2. Instrukcja użytkowania aplikacji

W celu uruchomienia naszej aplikacji należy wejść pod podany adres: <http://megumin.xlx.pl/>. Po wpisaniu adresu naszym oczom ukaże się strona główna:



Rysunek 91: Strona główna aplikacji

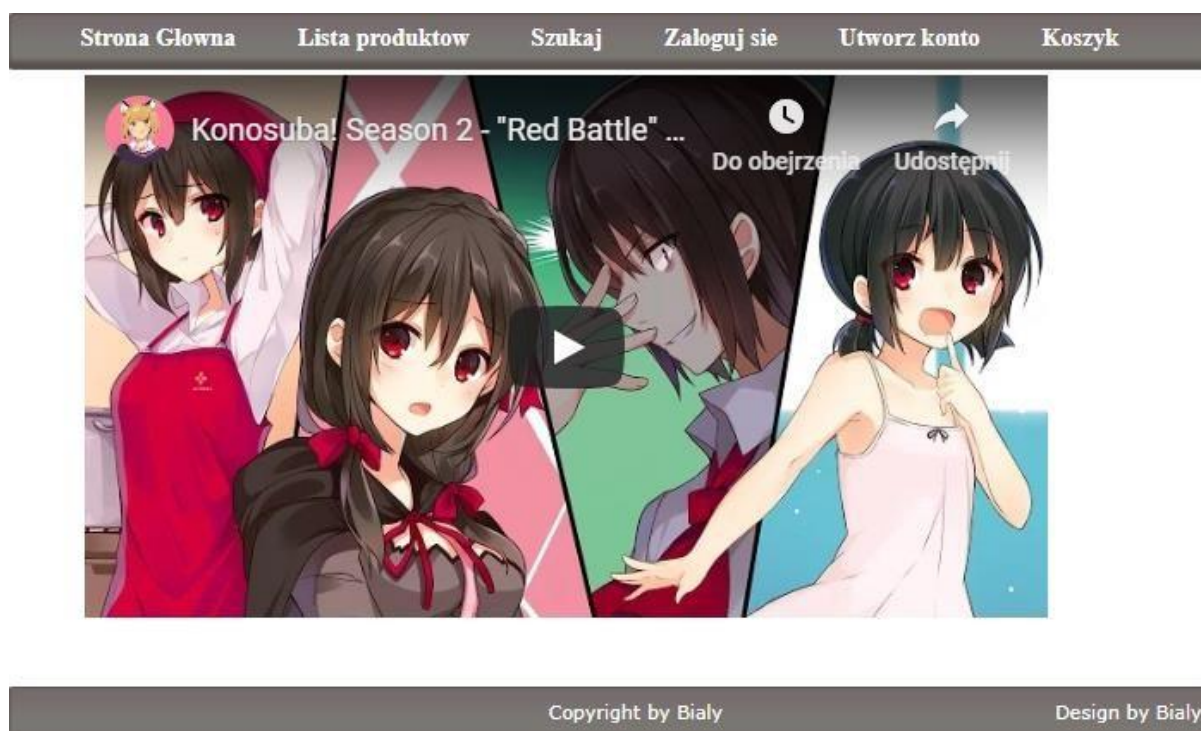
Tak jak wspominałem wcześniej serwis hostingowy po okresie próbnym wynoszącym jeden tydzień zaczął wyświetlać reklamy, które są dość impertynencko nakładane na górny pasek nawigacji. Dobrym rozwiązaniem jest włączenie ad block'a, gdyż nie mamy pewności co do treści wyświetlanych reklam.

Aplikacja posiada specjalny pasek nawigacyjny znajdujący się na górze, który znacznie ułatwia poruszanie się po stronie. Wszystkie przyciski oraz logo są interaktywne i przenoszą nas do wybranej destynacji. W zależności od preferencji użytkownika dostać się na stronę główną można na dwa, a tak właściwie to trzy sposoby (ale o tym w dalszej części instrukcji) poprzez naciśnięcie przycisku Strona główna bądź logo strony.



Rysunek 92: Logo strony

Dodatkowo pomyśleliśmy, aby zadbać o dobry humor użytkowników naszej aplikacji, dlatego umieściliśmy filmik z relaksującą muzyką, która wywoła uśmiech na twarzy.



Rysunek 93: Filmik relaksacyjny umieszczony na stronie

W celu wyświetlenia listy produktów znajdujących się w naszym sklepie należy kliknąć przycisk Lista produktów na pasku nawigacyjnym.

Strona Główna	Lista produktów	Szukaj	Zaloguj się	Utworz konto	Koszyk
<h2>Lista produktów</h2>					
Nazwa	Liczba	Dział	Cena netto		
Sonair	1	Dasyurus viverrinus	359.54		
Veribet	1	Amblyrhynchus crista	245.4		
Fintone	1	Zosterops pallidus	914.77		
Domainer	0	Gazella granti	761.22		
Aerified	0	Sciurus vulgaris	283.78		
It	1	Agkistrodon piscivor	164.28		
Kanlam	0	Otaria flavescens	397.71		
Span	0	Tragelaphus angasi	852.74		

Rysunek 94: przykładowy widok listy produktów

Wyszukiwanie interesujących nas produktów, które chcemy potem zakupić następuje po naciśnięciu przycisku Szukaj.

Strona Główna	Lista produktów	Szukaj	Zaloguj się	Utworz konto	Koszyk
<h2>Wyszukiwanie produktów</h2>					
Wyszukaj według: <input type="text" value="Nazwa"/>					
Szukane wyrażenie: <input type="text"/>					
<input type="button" value="Wyslij"/>					
<input type="button" value="Wróć"/>					
Copyright by Bialy					
Design by Bialy					

Rysunek 95: Wyszukiwarka produktów

Tutaj mamy możliwość wyszukania interesującego nas produktu po nazwie, liczbie dostępnej na magazynie, dziale oraz cenie netto. Przykładowe wyrażenie, które możemy wyszukać to Sonair. Po wpisaniu wyrażenia klikamy przycisk wyślij i powinna nam się ukazać lista znalezionych produktów. Dodatkowo pojawia się trzecia opcja powrotu do strony głównej bądź poprzedniej o której wspominałem wcześniej a mianowicie przycisk wrót.



Strona Główna Lista produktów Szukaj Zaloguj się Utwórz konto Koszyk

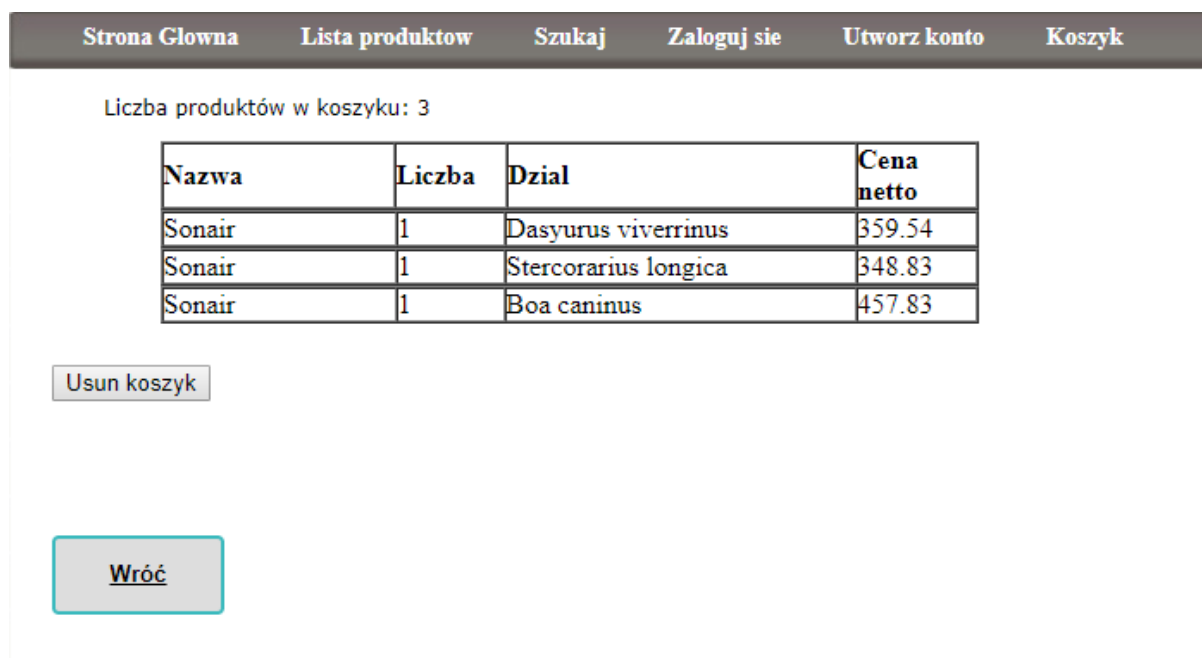
Wyniki wyszukiwania produktów

Liczba znalezionych produktów: 16

- Nazwa: Sonair
Liczba: 1
Dział: Dasyurus viverrinus
Cena netto: 359.54
- Nazwa: Sonair
Liczba: 0
Dział: Paraxerus cepapi
Cena netto: 292.87
- Nazwa: Sonair
Liczba: 0
Dział: Orcinus orca
Cena netto: 357.18

Rysunek 96: Przykładowy wynik wyszukiwania

Tutaj interesujące nas produkty można dodać do koszyka poprzez naciśnięcie guzika Dodaj do koszyka. Po dodaniu kilku produktów nasz koszyk powinien wyglądać następująco:



Strona Główna Lista produktów Szukaj Zaloguj się Utwórz konto Koszyk

Liczba produktów w koszyku: 3

Nazwa	Liczba	Dział	Cena netto
Sonair	1	Dasyurus viverrinus	359.54
Sonair	1	Stercorarius longica	348.83
Sonair	1	Boa caninus	457.83

Rysunek 97: Widok koszyka po dodaniu kilku produktów

Możemy również usunąć wcześniej utworzony koszyk poprzez naciśnięcie przycisku Usuń koszyk.



Rysunek 98: Widok po usunięciu koszyka

Kolejną opcją jaką mamy do wyboru jest zalogowanie się na wcześniej utworzone konto. Dokonuje się tego poprzez naciśnięcie przycisku Zaloguj się, potem należy podać prawidłowe dane oraz nacisnąć przycisk Zaloguj. Jeśli proces przebiegł poprawnie otrzymamy następujący komunikat:



Rysunek 99: Wyświetlany komunikat po zalogowaniu się

Po zalogowaniu się możemy zauważyć, iż nasz pasek nawigacyjny uległ drobnym zmianom, a mianowicie pojawiła się opcja Edytuj dane. Po wybraniu tej opcji zostanie wyświetlony panel z naszymi aktualnymi danymi osobowymi oraz pustymi polami, dzięki którym możemy wprowadzić edytowane dane. Naciśnięcie przycisku Zapisz zmiany spowoduje zapisanie z edytowanych danych, natomiast przycisk Usuń użytkownika spowoduje usunięcie naszego konta.

Strona Główna Lista produktów Szukaj Edytuj dane Wyloguj Koszyk

Dane osobowe

Twoje aktualne dane:

Nazwa firmy: Niech Bog ma nas w opiece
Nip:
Imie: A
Nazwisko: B
Nr telefonu: 6962138230
Adres e-mail: gadshed2@gov.uk

Edytuj dane:

Nazwa firmy:
Nip:
Imie:
Nazwisko:
Nr telefonu:
Adres e-mail:

Copyright by Bialy Design by Bialy

Rysunek 100: Widok po naciśnięciu przycisku Edytuj dane

Dane osobowe

Twoje aktualne dane:

Nazwa firmy: Niech Bog ma nas w opiece
Nip: 5432
Imie: A
Nazwisko: B
Nr telefonu: 6962138230
Adres e-mail: mailtestowy@gmail.com

Edytuj dane:

Nazwa firmy:
Nip:
Imie:
Nazwisko:
Nr telefonu:
Adres e-mail:

Rysunek 101: Widok po zmianie numeru Nip oraz adresu e-mail

Po wprowadzeniu zmian możemy się wylogować poprzez naciśnięcie przycisku Wyloguj.



Rysunek 102: Komunikat widoczny po wylogowaniu się

Kolejną opcją jest możliwość utworzenia konta w naszym serwisie poprzez naciśnięcie przycisku Utwórz konto. Wymagane dane do podania to login oraz hasło. Następnie należy nacisnąć przycisk Zarejestruj.

The registration form is displayed within a dark grey navigation bar at the top, which contains links: Strona Główna, Lista produktów, Szukaj, Zaloguj się, Utwórz konto, and Koszyk. The main heading is 'Rejestracja'. Below it are two input fields: 'login:' and 'hasło:'. A 'Zarejestruj' button is positioned below the password field. A 'Wróć' button is located further down. At the bottom of the page, a dark grey footer contains 'Copyright by Bialy' and 'Design by Bialy'.

Strona Główna	Lista produktów	Szukaj	Zaloguj się	Utwórz konto	Koszyk
---------------	-----------------	--------	-------------	--------------	--------

Rejestracja

login:

hasło:

Zarejestruj

Wróć

Copyright by Bialy Design by Bialy

Rysunek 103: Rejestracja

The registration result page features the same dark grey navigation bar at the top with links: Strona Główna, Lista produktów, Szukaj, Zaloguj się, Utwórz konto, and Koszyk. The main heading is 'Wynik rejestracji'. Below it, the text 'Liczba rekordów dodanych do bazy: 1' is shown. A 'Zaloguj się' button is located at the bottom. The dark grey footer at the bottom contains 'Copyright by Bialy' and 'Design by Bialy'.

Strona Główna	Lista produktów	Szukaj	Zaloguj się	Utwórz konto	Koszyk
---------------	-----------------	--------	-------------	--------------	--------

Wynik rejestracji

Liczba rekordów dodanych do bazy: 1

Zaloguj się

Copyright by Bialy Design by Bialy

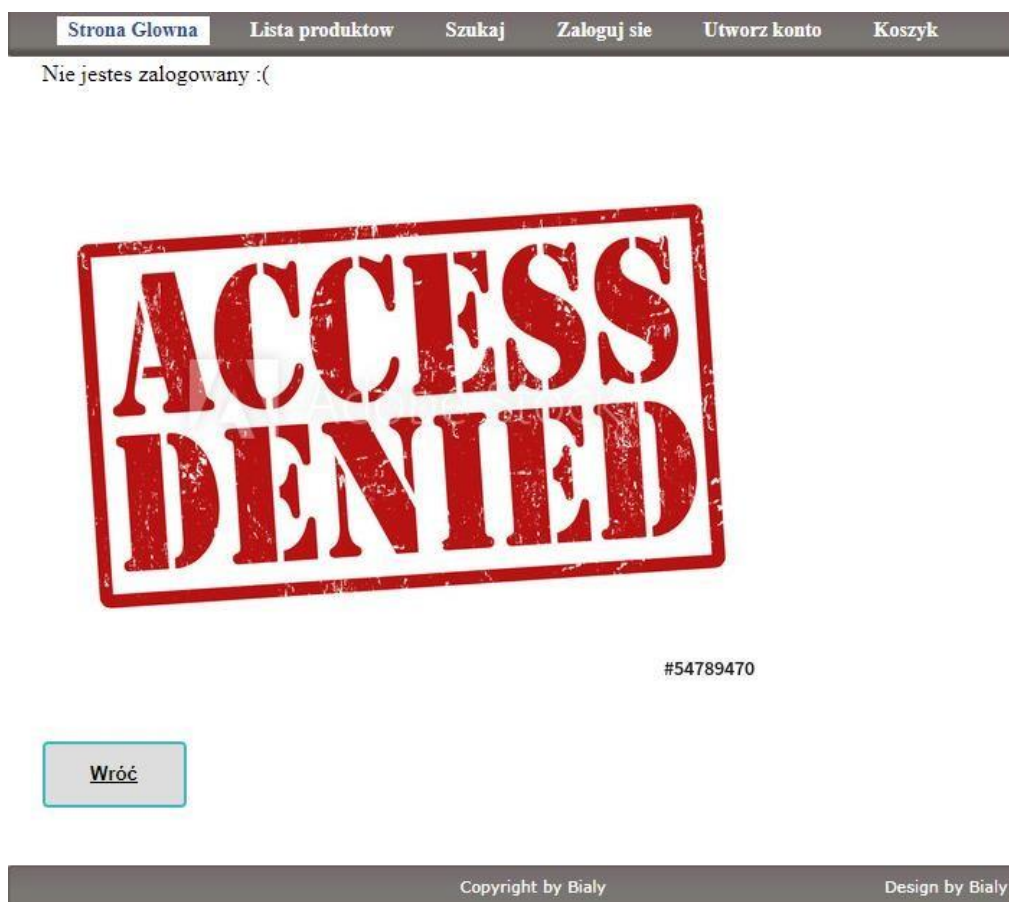
Rysunek 104: Wynik rejestracji

Dla ułatwienia mamy od razy dostępny przycisk Zaloguj się, która pozwala zalogować się na wcześniej utworzone konto.

Ostatnim funkcjonalnym przyciskiem na naszej aplikacji jest przycisk Koszyk, który pozwala w dowolnym momencie podejrzeć jego zawartość.

5.3. Testowanie opracowanych funkcji systemu

Pełna funkcjonalność naszej aplikacji została przetestowana podczas tworzenia punktu numer 3 co można zauważyć na zamieszczonych tam obrazkach. Dodatkową funkcjonalnością, o której nie wspomniałem podczas tamtego punktu jest blokowanie dostępu do danych dla niezalogowanych użytkowników. Po wpisaniu naszego adresu www.megumin.xlx.pl/dane.php niezalogowanemu użytkownikowi wyświetli się następujący komunikat:



Rysunek 105: Odmowa dostępu dla niezalogowanego użytkownika

5.4. Omówienie wybranych rozwiązań programistycznych

Aplikacja dostępowa została przez nas napisana przy wykorzystaniu takich technologii jak: PHP, HTML oraz CSS. Postawiliśmy na nie z tego względu, że istnieje do nich duża baza poradników wprowadzająca nowicjusza bezboleśnie w dany język krok po kroku. Dodatkowo jest to dość często polecane połączenie przeznaczone dla aplikacji webowych oraz część naszej grupy projektowej miała wcześniej styczność z PHP oraz językiem HTML co ułatwiło cały proces wdrażania się.

5.4.1. Implementacja interfejsu dostępu do bazy danych

Za dostęp do bazy danych odpowiadał nam funkcja `mysql`, która wymagała podania takich parametrów jak: nazwa użytkownika, nazwa serwera bądź jak w naszym przypadku jego IP, hasło do bazy danych oraz nazwę bazy, z którą próbujemy nawiązać połączenie:

```
$mysql=mysqli_connect('localhost', 'root', 'root');  
//połączenie z MySQL  
if (!$mysql)  
{  
    echo 'Brak polaczenia z MySQL!';  
}  
else{  
    $baza=mysqli_select_db($mysql, 'test');  
    //wybór potrzebnej bazy  
    if (!$baza)  
    {  
        echo 'Brak polaczenia z baza uzytkownikow!';  
    }  
}
```

Rysunek 106: Interfejs dostępu do bazy danych

5.4.2. Implementacja wybranych funkcjonalności systemu

- a) Implementacja funkcji wyświetlenia listy produktów polegała głównie na nawiązaniu połączenia z bazą MySQL oraz stworzenia ładnie wyglądającej tabelki w HTMLu, gdzie owa lista będzie się znajdować. Z racji tego, iż korzystamy z wersji PHP powyżej 7.0 wykorzystane przez nas funkcję do nawiązania połączenia z bazą danych różnią się od tych wykorzystywanych w wersjach poprzednich. Niestety PHP nie wspiera wstecznej kompatybilności, dlatego przykładowo połączenie nawiązuje się poprzez funkcję `mysqli` zamiast `mysql`. Z racji bezpieczeństwa dane dostępne zostały zamazane.

```

<?php
$servername = " ";
$username = " ";
$password = " ";
$dbname = " ";

// Create connection
$mysqli = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}

```

Rysunek 107: Inny sposób nawiązania połączenia z bazą danych w języku PHP

W polu dla zmiennej \$servername wpisujemy adres ip, na którym działa nasza aplikacja. Pole \$username wypełniamy nazwą użytkownika jaka została przydzielona nam w serwisie, \$password uzupełniamy wygenerowanym hasłem natomiast \$dbname oznacza nazwę bazy danych, z którą chcemy nawiązać połączenie. Następnie tworzymy połączenie z bazą danych wykorzystując wcześniej wspomnianą funkcję mysqli. W razie wystąpienia błędu zostanie nam zwrócony komunikat „Connection failed”. Kolejnym krokiem jest przesłanie zapytanie SQL naszej bazie, która potem zwróci nam otrzymane rezultaty. Dodatkowo wewnątrz PHP został wykorzystany HTML w celu wyświetlenia rezultatów w przyjemniejszej dla oka tabeli.

Za przesłanie zapytania SQL opowiada funkcja mysqli_query(), natomiast wyświetlenie otrzymanych rezultatów odbywa się przy pomocy funkcji mysqli_fetch_assoc().

```

$results=mysqli_query($mysqli, "SELECT * FROM produkt");
echo "<center><h1><b>Lista produktow</b></h1></center>";
echo "<center><table width='475' border='1' cellpadding='0' cellspacing='0'>
    <tr><td width='100'><b>Nazwa</b></td><td width='45'><b>Liczba</b></td><td width='150'>
    <b>Dzial</b></td><td width='50'><b>Cena netto</b></td></tr>
</table></center>";
while ($txt = mysqli_fetch_assoc($results)){
    echo "<center><table width='475' border='1' cellpadding='0' cellspacing='0'>
    <tr><td width='100'>".$txt['nazwa']. "</td><td width='45'>".$txt['liczba']. "</td>
    <td width='150'>".$txt['dzial']. "</td><td width='50'>".$txt['cena_netto']. "</td></tr>
    </table></center>";
}

```

Rysunek 108: Przesłanie zapytania SQL do bazy oraz wyświetlenie rezultatu

- b) Implementacja funkcji wyświetlenia oraz edycji danych dostępnych po zalogowaniu się na swoje konto polegała w dużej mierze na przekazywaniu sobie zmiennych pomiędzy skryptami. Na samym początku należy nawiązać połączenie z bazą danych. Cała procedura jest opisana powyżej. Kolejnym krokiem jest sprawdzenie klucza obcego dla podanego loginu oraz hasła.

```
$zapytanie="select dane_osobowe_id_dane_osobowe from uzytkownik where login='$login2' and haslo='$haslo2'";  
$wynik=mysqli_query($mysql,$zapytanie);  
$wiersz = $wynik->fetch_assoc();
```

Rysunek 109: Odczyt id klucza obcego dla podanego loginu oraz hasła

Następnie na podstawie tego klucza odczytujemy dane tej osoby:

```
$id3 = stripslashes($wiersz['dane_osobowe_id_dane_osobowe']);  
$idl="select * from dane_osobowe where id_dane_osobowe='$id3'";  
$rezultat=mysqli_query($mysql,$idl);  
$wierszl = $rezultat->fetch_assoc();
```

Rysunek 110: Odczyt danych osobowych z bazy dla podanego id klucza obcego

Za wyświetlenie danych dla odpowiedniej osoby odpowiada na połączenie HTML wraz z PHP.

```
echo '<h1>Dane osobowe</h1><br />';  
echo '<h2>Twoje aktualne dane:</h2><br />';  
echo 'Nazwa firmy: '.stripslashes($wierszl['nazwa_firmy']).'<br />';  
echo 'Nip: '.stripslashes($wierszl['nip']).'<br />';  
echo 'Imie: '.stripslashes($wierszl['imie']).'<br />';  
echo 'Nazwisko: '.stripslashes($wierszl['nazwisko']).'<br />';  
echo 'Nr telefonu: '.stripslashes($wierszl['nr_telefonu']).'<br />';  
echo 'Adres e-mail: '.stripslashes($wierszl['email']).'<br />';  
echo '<br /> <br />';
```

Rysunek 111: Wyświetlenie aktualnych danych osobowych

Aktualizacja danych odbywa się przy pomocy warunku if, który sprawdza czy pole, do którego możemy wpisać edytowane dane jest puste oraz przy pomocy zapytania SQL:

```
if(!empty($_POST['nazwa_firmy1']))
{
    $nazwa_firmy1 = $_POST['nazwa_firmy1'];
    $sql = "UPDATE dane_osobowe SET nazwa_firmy='$nazwa_firmy1' WHERE id_dane_osobowe='$id3'";
    $wynik2 = mysqli_query($mysql,$sql);
}

if(!empty($_POST['nip1']))
{
    $nip1 = $_POST['nip1'];
    $sql3 = "UPDATE dane_osobowe SET nip='$nip1' WHERE id_dane_osobowe='$id3'";
    $wynik3 = mysqli_query($mysql,$sql3);
}

if(!empty($_POST['imiel']))
{
    $imiel = $_POST['imiel'];
    $sql4 = "UPDATE dane_osobowe SET imie='$imiel' WHERE id_dane_osobowe='$id3'";
    $wynik4 = mysqli_query($mysql,$sql4);
}
```

Rysunek 112: Edycja danych przy pomocy języka PHP

Realizacja wyżej wymienionych pól tekstowych oraz dwóch przycisków odpowiadających za zapisanie wprowadzonych zmian oraz usunięcie użytkownika odbywa się przy wykorzystaniu HTMLa, gdzie type="test" oznacza puste pole tekstowe natomiast type="submit" oznacza guzik, który przeprowadza akcję dopiero po wciśnięciu go.

```
<h2>Edytuj dane:</h2><br />
<form action="dane.php" method="post">
    Nazwa firmy: <input type="text" name="nazwa_firmy1"/> <br />
    Nip: <input type="text" name="nip1" /> <br />
    Imie: <input type="text" name="imiel" /> <br />
    Nazwisko: <input type="text" name="nazwisko1" /> <br />
    Nr telefonu: <input type="text" name="nr_telefonu1" /> <br />
    Adres e-mail: <input type="text" name="email1" /> <br />
    <input type="submit" name="submit" value="Zapisz zmiany" /> <br />
    <input type="submit" name="submit1" value="Usun uzytkownika" />
</form>
```

Rysunek 113: Puste pola przeznaczone do edycji danych w języku HTML

5.4.3. Implementacja mechanizmów bezpieczeństwa

Pierwszym mechanizmem bezpieczeństwa, który wdrożyliśmy jest oczywiście konieczność podania poprawnego loginu oraz hasła, aby móc zobaczyć bądź z edytować dane osobowe. Próba dostępu bez spełnienia tych wymagań kończy się komunikatem o odmowie dostępu, co zostało opisane w punkcie 4. Kolejnym mechanizmem bezpieczeństwa była nieudolna próba zaszyfrowania hasła. Po przeszukaniu przeróżnych forów nadal nie znaleźliśmy rozwiązania aby funkcja `sha1()` nie powodowała błędów.

```
//$haslo=sha1($haslo); //zaszyfrowanie hasła
```

Rysunek 114: Próba zaszyfrowania hasła w PHP

6. Podsumowanie i wnioski

Reasumując nasze wysiłki udało nam się stworzyć praktycznie działający sklep internetowy postawiony na swojej własnej domenie. Spełniliśmy wszystkie nasze założenia i cele, a nawet wykroczyliśmy znacznie poza nie tworząc oprawę graficzną wcss, tworząc dodatkowe schematy UML, programując ułatwiające widoki tabel, wykonując dodatkowe testy bazy danych oraz wynajmując domenę internetową zamiast posługiwania się host'em lokalnym. W szczególności opracowaliśmy założenia:

- Sformułowaliśmy założenia projektowe oraz wymagania funkcjonalne oraz нефункционалне. Przygotowaliśmy dokumentację omawiającą te zagadnienia.
- Zaprojektowaliśmy schematy UML oraz schematy fizyczno-logiczne. Skonstruowaliśmy dokumentację opisującą ich logikę.
- Stworzyliśmy bazę danych korzystając z środowiska MySQL. Uruchomiliśmy ją korzystając z PHPMyAdmin na lokalnym urządzeniu. Przygotowaliśmy dokumentację opisującą ten fragment projektu. Skonstruowaliśmy przykładowe mock-up'y aplikacji dostępowej.
- Opracowaliśmy aplikację dostępową korzystając z PHP. Wykonaliśmy dokumentację by wskazać kierunek konstruowania podobnej bazy przez inną osobę.

Literatura

W naszej pracy korzystaliśmy głównie z Internetu – stron filmików na YouTube oraz wykładów m.in. dr inż. Romana Ptaka. Oto spis stron, z których korzystaliśmy:

- <http://staff.uz.zgora.pl/afiedoro/pliki/modelowanie1.pdf>
- http://roman.ptak.staff.iar.pwr.wroc.pl/BD_wyklad_nr1_ver7.pdf
- http://roman.ptak.staff.iar.pwr.wroc.pl/BD_wyklad_nr2_ver6.pdf
- http://roman.ptak.staff.iar.pwr.wroc.pl/BD_wyklad_nr3_ver6.pdf
- https://4programmers.net/Forum/Inzynieria_oprogramowania/211611-przypadki_uzycia_dla_sklepu_internetowego
- http://www.obrazki.wiedzanaplus.pl/programowanie/uml/projekt_sklep/use_case_diagram/diagram_przypadkow_uzycia_sklep_komputerowy_sprzedaz_uml.jpg
- <http://staruml.io/>
- <http://dia-installer.de/>
- <https://www.wiedzanaplus.pl/programowanie/33-uml/68-jezyk-uml-projekt-sklepu-komputerowego.html>
- <https://www.youtube.com/watch?v=V7Ptd3BTiV0>
- <https://docplayer.pl/243947-Diagramy-przypadkow-uzycia-ms-visio.html>
- <https://app.moqups.com/egBbOIOnC3/view>
- https://www.michalwolski.pl/wp-content/uploads/2012/03/najczesciej_stosowana_notacja_UML_2011.html_m55cd26a7_thumb.gif
- <http://jsystems.pl/blog/artykul.html?id=253>
- <http://generator danych.testery.pl/>
- <http://baza-danych.blogspot.com/2011/11/mysql-rand.html>
- <https://www.mockaroo.com/>
- <https://apex.oracle.com/en/>
- https://forumapex.dbep.org/ords/f?p=FORUM_APEX:TEMAT:0::::URL_FRIENDLY,MES_ID,CAT_ID:MqSql-Workbench---eksport-tabel-do-APEX,2421,5
- <https://www.youtube.com/watch?v=Esyen1NXsbq>
- <https://opensolution.org/system-cms-quick-cms.html>
- <https://www.youtube.com/watch?v=IW3vKHWJlY>
- <https://apexutil.blogspot.com/2018/07/oracle-apex-with-mysql.html>
- <https://pl.wordpress.org/>
- <https://jekyllrb.com/docs/>
- <https://cakephp.org/>
- <https://github.com/chrisveness/koa-sample-web-app-api-mysql>
- <https://github.com/edlangley/inventory-webapp>
- <https://www.youtube.com/watch?v=xxaUVK-miwg>
- <https://www.youtube.com/watch?v=JQ2Z1XJ7qT0>
- <https://forum.pasja-informatyki.pl/17152/jak-podpiac-php-do-html>
- <https://www.forumweb.pl/php/tabela-w-php,39529>
- <https://forum.webhelp.pl/pocztakujacy-webmaster/php-html-tabela-t244876.html>
- http://www.teacher.webd.pl/kurs_php/k_10_0_0.php
- http://www.teacher.webd.pl/kurs_php/k_11_0_0.php

- http://www.teacher.webd.pl/kurs_php/k_12_0_0.php
- <http://www.php.pl/Wortal/Artykuly/PHP/Podstawy/Tutorial-PHP-MySQL/Aktualizacja-i-usuwanie-danych>
- <https://forum.pasja-informatyki.pl/164/wywołanie-funkcji-php-przyciskiem>
- <http://webmaster.helion.pl/index.php/kursjs-obsluga-zdarzen-i-elementow-strony/kursjs-obsluga-formularzy/kursjs-obsluga-elementow-formularza>
- <https://pl.000webhost.com/darmowa-rejestracja>
- <https://www.prv.pl/hosting/>
- <https://www.prv.pl/pomoc-faq/25,gdzie-znajde-dane-potrzebne-do-skonfigurowania-polaczenia-z-baza-danych>
- <https://www.prv.pl/pomoc-faq/95,jak-wykonac-export-zawartosci-sql-w-phpmyadmin>
- <https://pomoc.premium.pl/1544/jak-przeniesc-baze-danych-do-premium-pl/>
- <https://zemez.io/wordpress/support/how-to/no-database-selected-error-importing-sql-files/>
- <https://hatchful.shopify.com/>

Spis rysunków

Rysunek 1: Model konceptualny.....	6
Rysunek 2: Model logiczno-fizyczny.....	7
Rysunek 3: Widok produktu	7
Rysunek 4: Widok transakcji	7
Rysunek 5: Widok klienta	8
Rysunek 6: Diagram przypadków użycia UML	8
Rysunek 7: Diagram klas UML	9
Rysunek 8: Diagram czynności UML – logowanie	10
Rysunek 9: Diagram czynności UML – administrator	11
Rysunek 10: Diagram czynności UML - edycja danych osobowych lub adresu	11
Rysunek 11: Diagram czynności UML - przeglądanie produktów.....	12
Rysunek 12: Diagram czynności UML – płatność.....	12
Rysunek 13: Mock-up rejestracji.....	15
Rysunek 14: Mock-up edycji danych osobowych	15
Rysunek 15: Mock-up wyszukiwania produktu.....	16
Rysunek 16: Mock-up produktu.....	16
Rysunek 17: Mock-up koszyka.....	17
Rysunek 18: Mock-up płatności.....	17
Rysunek 19: Implementacja bazy danych krok	18
Rysunek 20: Implementacja bazy danych krok 2 – korzystanie z funkcji forward engineer	19
Rysunek 21: Implementacja bazy danych krok 3 – wybór parametrów	19
Rysunek 22: Implementacja bazy danych krok 4 – logowanie się jako root serwera MySQL	20
Rysunek 23: Implementacja bazy danych krok 5 - wybór elementów do eksportu	20
Rysunek 24: Komunikat o poprawnej implementacji bazy danych	21
Rysunek 25: Wygenerowany fragment skryptu przez program - tworzenie tabeli adres	22
Rysunek 26: Wygenerowany fragment skryptu przez program - tworzenie tabeli pośredniej	22
Rysunek 27: Wygenerowany fragment skryptu przez program - tworzenie widoku klienta	23
Rysunek 28: Utworzenie konta administratora i nadanie mu uprawnień	23
Rysunek 29: Utworzenia konta gościa i nadanie mu uprawnień	23
Rysunek 30: Utworzenie konta klient i nadanie mu uprawnień	24
Rysunek 31: Utworzenie konta serwis transakcyjny i nadanie mu uprawnień	24
Rysunek 32: Weryfikacja poprawności skryptu	24
Rysunek 33: Weryfikacja nadanych uprawnień przy wykorzystaniu terminala	25
Rysunek 34: Logowanie się jako administrator	25
Rysunek 35: Sprawdzenie uprawnień Administratora dla tabeli „vat”	26
Rysunek 36: Weryfikacja poprawności skryptu dla testu na TAK	26
Rysunek 37: Stan tabeli po dodaniu nowej stawki vat.....	26
Rysunek 38: Stan tabeli po edycji stawki vat z indeksem 1	26
Rysunek 39: Stan tabeli po usunięciu stawki vat z indeksem 1	26
Rysunek 40: Przeprowadzanie testu uprawnień dla administratora dla tabeli "kategoria"	27
Rysunek 41: Weryfikacja poprawności skryptu dla testu na TAK	27
Rysunek 42: Stan tabeli po dodaniu nowej kategorii.....	27
Rysunek 43: Stan tabeli po edycji kategorii z indeksem 1	27
Rysunek 44: Stan tabeli po usunięciu kategorii z indeksem 1	27
Rysunek 45: Logowanie się jako gość	28

Rysunek 46: Sprawdzenie uprawnień Gościa dla tabeli „vat”	29
Rysunek 47: Ukazanie weryfikacji testu na TAK i na NIE	29
Rysunek 48: Stan tabeli „vat” przed	29
Rysunek 49: Stan tabeli „vat” po	29
Rysunek 50: Sprawdzenie uprawnień Gościa dla tabeli „kategoria”	30
Rysunek 51: Weryfikacja testu na TAK i na NIE	30
Rysunek 52: Stan tabeli „kategoria” przed	30
Rysunek 53: Stan tabeli „kategoria” po	30
Rysunek 54: Logowanie się jako serwis transakcyjny	31
Rysunek 55: Sprawdzenie uprawnień Serwisu transakcyjnego dla tabeli „vat”	32
Rysunek 56: Weryfikacja testu na TAK	32
Rysunek 57: Stan tabeli „vat” przed	32
Rysunek 58: Stan tabeli „vat” po	32
Rysunek 59: Sprawdzenie uprawnień Serwisu transakcyjnego dla tabeli „płatność”	33
Rysunek 60: Weryfikacja testu na TAK i na NIE	33
Rysunek 61: Stan tabeli „płatność” przed	33
Rysunek 62: Stan tabeli „płatność” po	33
Rysunek 63: Logowanie się jako klient	34
Rysunek 64: Sprawdzenie uprawnień Serwisu transakcyjnego dla tabeli „produkt”	34
Rysunek 65: Weryfikacja testu na TAK i na NIE	34
Rysunek 66: Stan tabeli „produkt” przed	35
Rysunek 67: Stan tabeli „produkt” po	35
Rysunek 68: Sprawdzenie uprawnień Klienta dla tabeli „koszyk”	36
Rysunek 69: Weryfikacja testu na TAK i na NIE	36
Rysunek 70: Stan tabeli „koszyk” przed	36
Rysunek 71: Stan tabeli „koszyk” po	36
Rysunek 72: Dane osobowe wprowadzone losowo	37
Rysunek 73: Panel kontrolny pakietu XAMPP	38
Rysunek 74: Widok panelu kontrolnego po uruchomieniu interesujących nas modułów	39
Rysunek 75: Ustawienie parametrów w MySQL Workbench	40
Rysunek 76: Błędna generacja indexów przez Workbench'a	40
Rysunek 77: Pomyślne zakończenie się procesu Forward Engineer	41
Rysunek 78: Panel kontrolny XAMPPa - przejście do zarządzania bazą	41
Rysunek 29: Widok bazy danych z poziomu phpmyadmin	42
Rysunek 30: Usunięcie ograniczeń kluczy obcych	42
Rysunek 81: Uruchamianie serwera Apache	43
Rysunek 82: Ustawianie hasła dla bazy MySQL	43
Rysunek 83: Konfiguracja hasła z poziomu panelu kontrolnego	43
Rysunek 84: Ustawienie hasła – plik konfiguracyjny	44
Rysunek 85: Panel administratora na stronie hostingowej	45
Rysunek 86: Eksportowanie bazy danych z serwera lokalnego	45
Rysunek 87: Import bazy danych na serwer hostingowy	46
Rysunek 88: Widok zaimportowanych tabeli	47
Rysunek 89: Zarządzanie skryptami	47
Rysunek 90: Dodawanie skryptów w języku php	48
Rysunek 91: Strona główna aplikacji	49
Rysunek 92: Logo strony	50
Rysunek 93: Filmik relaksacyjny umieszczony na stronie	50

Rysunek 94: przykładowy widok listy produktów	51
Rysunek 95: Wyszukiwarka produktów	51
Rysunek 96: Przykładowy wynik wyszukiwania	52
Rysunek 97: Widok koszyka po dodaniu kilku produktów	52
Rysunek 98: Widok po usunięciu koszyka	53
Rysunek 99: Wyświetlany komunikat po zalogowaniu się	53
Rysunek 100: Widok po naciśnięciu przycisku Edytuj dane	54
Rysunek 101: Widok po zmianie numeru Nip oraz adresu e-mail	55
Rysunek 102: Komunikat widoczny po wylogowaniu się	55
Rysunek 103: Rejestracja	56
Rysunek 104: Wynik rejestracji	56
Rysunek 105: Odmowa dostępu dla niezalogowanego użytkownika	57
Rysunek 106: Interfejs dostępu do bazy danych	58
Rysunek 107: Inny sposób nawiązania połączenia z bazą danych w języku PHP	59
Rysunek 108: Przesłanie zapytania SQL do bazy oraz wyświetlenie rezultatu	59
Rysunek 109: Odczyt id klucza obcego dla podanego loginu oraz hasła	60
Rysunek 110: Odczyt danych osobowych z bazy dla podanego id klucza obcego	60
Rysunek 111: Wyświetlenie aktualnych danych osobowych	60
Rysunek 112: Edycja danych przy pomocy języka PHP	61
Rysunek 113: Puste pola przeznaczone do edycji danych w języku HTML	61
Rysunek 114: Próba zaszyfrowania hasła w PHP	62

Spis tabel

1 Tabela uprawnień	13
2 Wyniki testów uprawnień dla administratora	28
3 Wyniki testów uprawnień dla gościa	31
4 Wyniki testów uprawnień dla serwisu transakcyjnego	34
5 Wyniki testów uprawnień dla serwisu klienta	37