

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

## PROJEKT Z BAZ DANYCH

### **Friendly kitchen**

Termin zajęć: Poniedziałek, 9:15–11:00

AUTOR/AUTORZY:

Agnieszka Płoszaj

Indeks: 218353

E-mail: 218353@student.pwr.edu.pl

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 .....	5
2.3. Wymagania niefunkcjonalne .....	5
2.3.1. Wykorzystywane technologie i narzędzia .....	5
2.3.2. Wymagania dotyczące rozmiaru bazy danych.....	5
2.3.3. Wymagania dotyczące bezpieczeństwa systemu .....	5
3. Projekt systemu .....	6
3.1. Projekt bazy danych.....	6
3.1.1. Analiza rzeczywistości i uproszczony model konceptualny.....	6
3.1.2. Model logiczny i normalizacja.....	7
3.1.3. Model fizyczny i ograniczenia integralności danych.....	7
3.1.4. Inne elementy schematu – mechanizmy przetwarzania danych .....	7
3.1.5. Projekt mechanizmów bezpieczeństwa na poziomie bazy danych.....	7
3.2. Projekt aplikacji użytkownika .....	8
3.2.1. Architektura aplikacji i diagramy projektowe .....	8
3.2.2. Interfejs graficzny i struktura menu .....	9
3.2.3. Projekt wybranych funkcji systemu .....	9
3.2.4. Metoda podłączania do bazy danych – integracja z bazą danych.....	10
3.2.5. Projekt zabezpieczeń na poziomie aplikacji .....	10
4. Implementacja systemu baz danych .....	11
4.1. Tworzenie tabel i definiowanie ograniczeń.....	11

4.2. Implementacja mechanizmów przetwarzania danych .....	11
4.3. Implementacja uprawnień i innych zabezpieczeń .....	12
4.4. Testowanie bazy danych na przykładowych danych.....	12
5. Implementacja i testy aplikacji.....	15
5.1. Instalacja i konfigurowanie systemu .....	15
5.2. Instrukcja użytkowania aplikacji .....	16
5.3. Testowanie opracowanych funkcji systemu .....	16
5.4. Omówienie wybranych rozwiązań programistycznych.....	17
5.4.1. Implementacja interfejsu dostępu do bazy danych .....	17
5.4.2. Implementacja wybranych funkcjonalności systemu .....	18
5.4.3. Implementacja mechanizmów bezpieczeństwa .....	20
6. Podsumowanie i wnioski.....	21
Literatura .....	21

# 1. Wstęp

## 1.1. Cel projektu

Celem projektu jest zaprojektowanie i implementacja bazy danych, która będzie częścią systemu pomagającego utrzymać porządek w kuchni, dzięki orientacji w rozlokowaniu produktów w niej się znajdujących, oraz wygodnym dostępie do kuchennych przepisów.

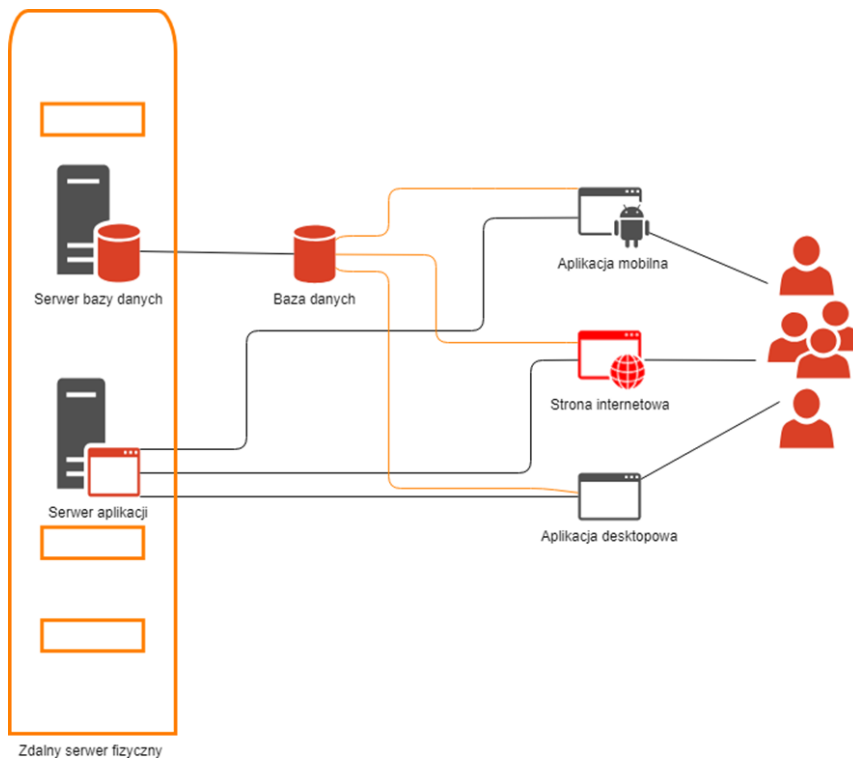
## 1.2. Zakres projektu

Zakres projektu można podzielić na:

1. Stworzenie bazy danych przechowującej informacje o produktach, składnikach oraz przepisach.
2. Stworzenie strony internetowej dla użytkowników pozwalającej na zarządzanie systemem.
3. Opracowanie dokumentacji całego projektu.

# 2. Analiza wymagań

## 2.1. Opis działania i schemat logiczny systemu



Rys.1 Schemat logiczny systemu.

Baza danych hostowana jest na serwerze i przechowuje wszystkie informacje o produktach oraz przepisach.

Strona internetowa jest również hostowana na serwerze.

Za pomocą strony internetowej, która łączy się z bazą danych, użytkownik zarządza systemem.

## 2.2. Wymagania funkcjonalne

Zalogowany użytkownik może :

- dodawać dane w ramach tabeli „Produkty”;
- usuwać dane w ramach tabeli „Produkty”;
- edytować dane w ramach tabeli „Produkty”;
- wyszukiwać dane w ramach tabeli „Produkty”;

Administrator może:

- dodawać dane w ramach tabeli „Produkty”;
- usuwać dane w ramach tabeli „Produkty”;
- edytować dane w ramach tabeli „Produkty”;
- wyszukiwać dane w ramach tabeli „Produkty”;
- dodawać dane w ramach tabeli „Składniki”;
- usuwać dane w ramach tabeli „Składniki”;
- edytować dane w ramach tabeli „Składniki”;
- wyszukiwać dane w ramach tabeli „Składniki”;
- dodawać dane w ramach tabeli „Przepisy”;
- usuwać dane w ramach tabeli „Przepisy”;
- edytować dane w ramach tabeli „Przepisy”;
- wyszukiwać dane w ramach tabeli „Przepisy”;
- przeglądać katalog za pośrednictwem strony internetowej.

## 2.3. Wymagania нефunkcjonalne

### 2.3.1. Wykorzystywane technologie i narzędzia

Baza danych zaprojektowana w MySQL.

Do komunikacji z bazą danych wykorzystywane będą standardowe zapytania SQL.

Strona internetowa napisana przy pomocy HTML5, CSS3 oraz PHP.

Do implementacji aplikacji wykorzystane zostanie środowisko Visual Studio 2017.

### 2.3.2. Wymagania dotyczące rozmiaru bazy danych

Wielkość bazy danych nie powinna przekroczyć: 500 instancji encji „Przepisy”, 1000 instancji encji „Produkty” oraz 5 000 instancji encji „Składniki”.

Z bazy danych jednocześnie korzystać może maksymalnie 5 osób.

### 2.3.3. Wymagania dotyczące bezpieczeństwa systemu

Z systemu korzystać mogą tylko zalogowani użytkownicy.

Wszyscy użytkownicy używają tego samego loginu oraz hasła.

Zarządzanie serwerem będzie możliwe tylko przez administratora (który będzie miał dostęp do kopii zapasowej).

Z racji nie przechowywania żadnych wrażliwych danych nie przewiduje się dodatkowej ochrony systemu.

### 3. Projekt systemu

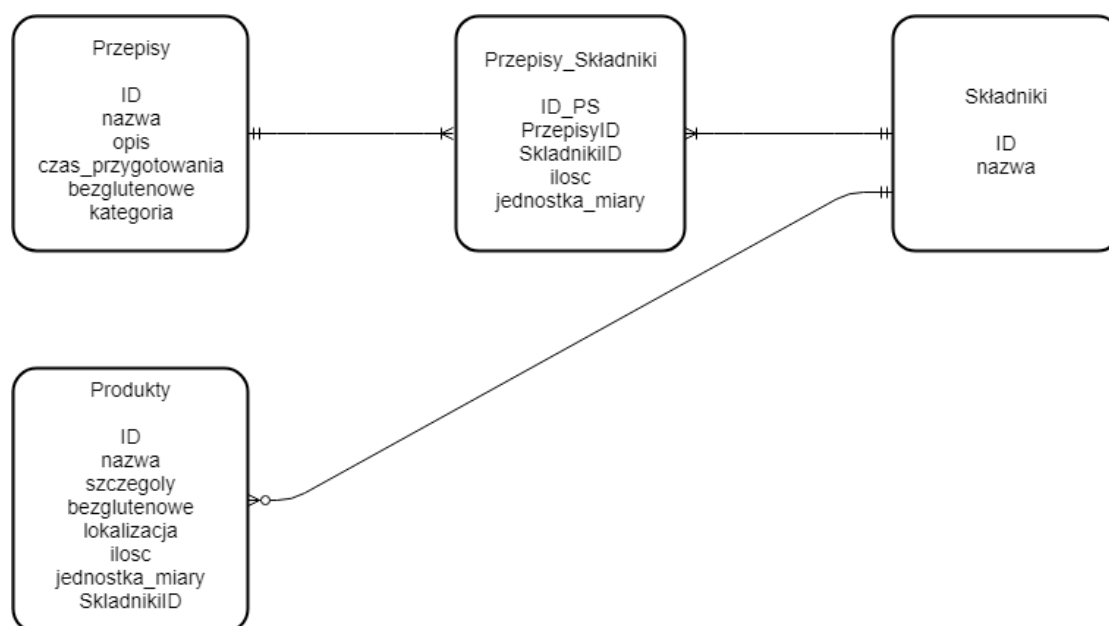
#### 3.1. Projekt bazy danych

##### 3.1.1. Analiza rzeczywistości i uproszczony model konceptualny



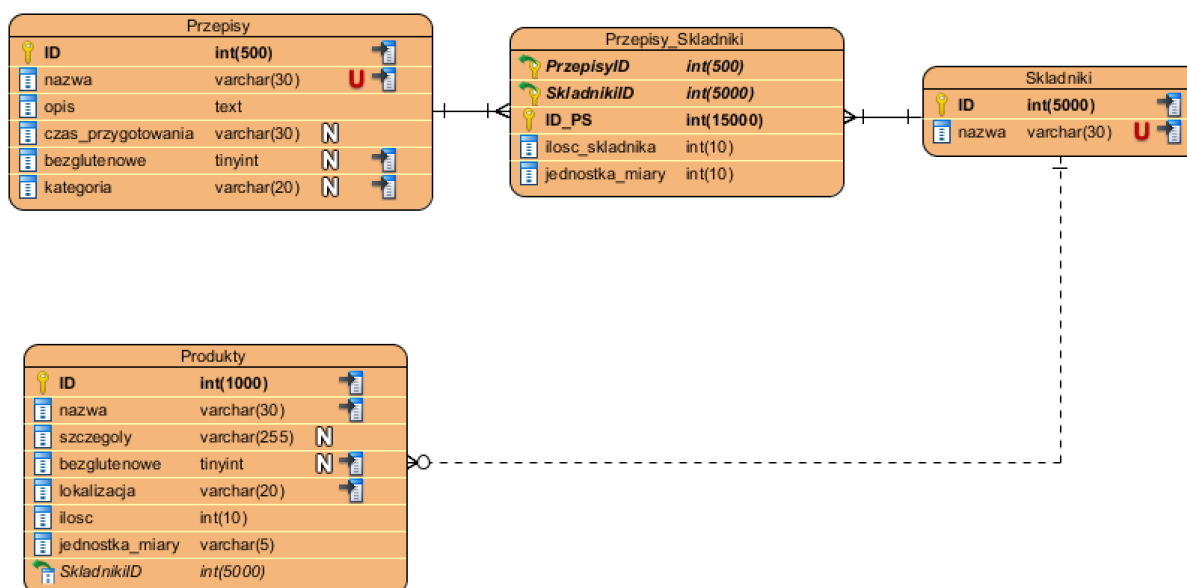
Rys.2 Uproszczony model konceptualny.

##### 3.1.2. Model logiczny i normalizacja



Rys.3 Model logiczny

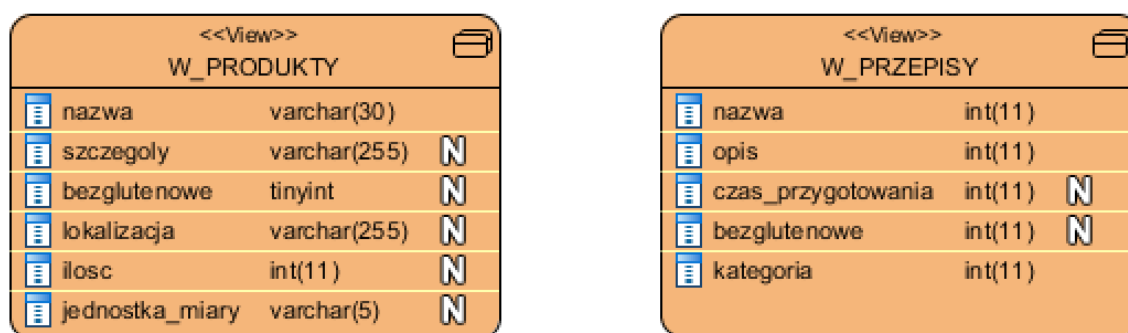
### 3.1.3. Model fizyczny i ograniczenia integralności danych



Rys.4 Model fizyczny

### 3.1.4. Inne elementy schematu – mechanizmy przetwarzania danych

W ramach aplikacji baza danych będzie odpowiedzialna za przechowywanie, edytowanie oraz wyszukiwanie danych. Reszta funkcjonalności będzie zapewniona przez odpowiednią implementację aplikacji użytkownika. Aby ułatwić i przyspieszyć pobieranie szczególnych przypadków danych baza danych posiada zaimplementowane następujące widoki: W\_przepisy, W\_skladniki oraz W\_produkty. Niestety w projekcie nie mogą wykorzystać triggerów, ponieważ serwer na którym tworzę całą bazę danych, nie pozwala mi na takie operacje.



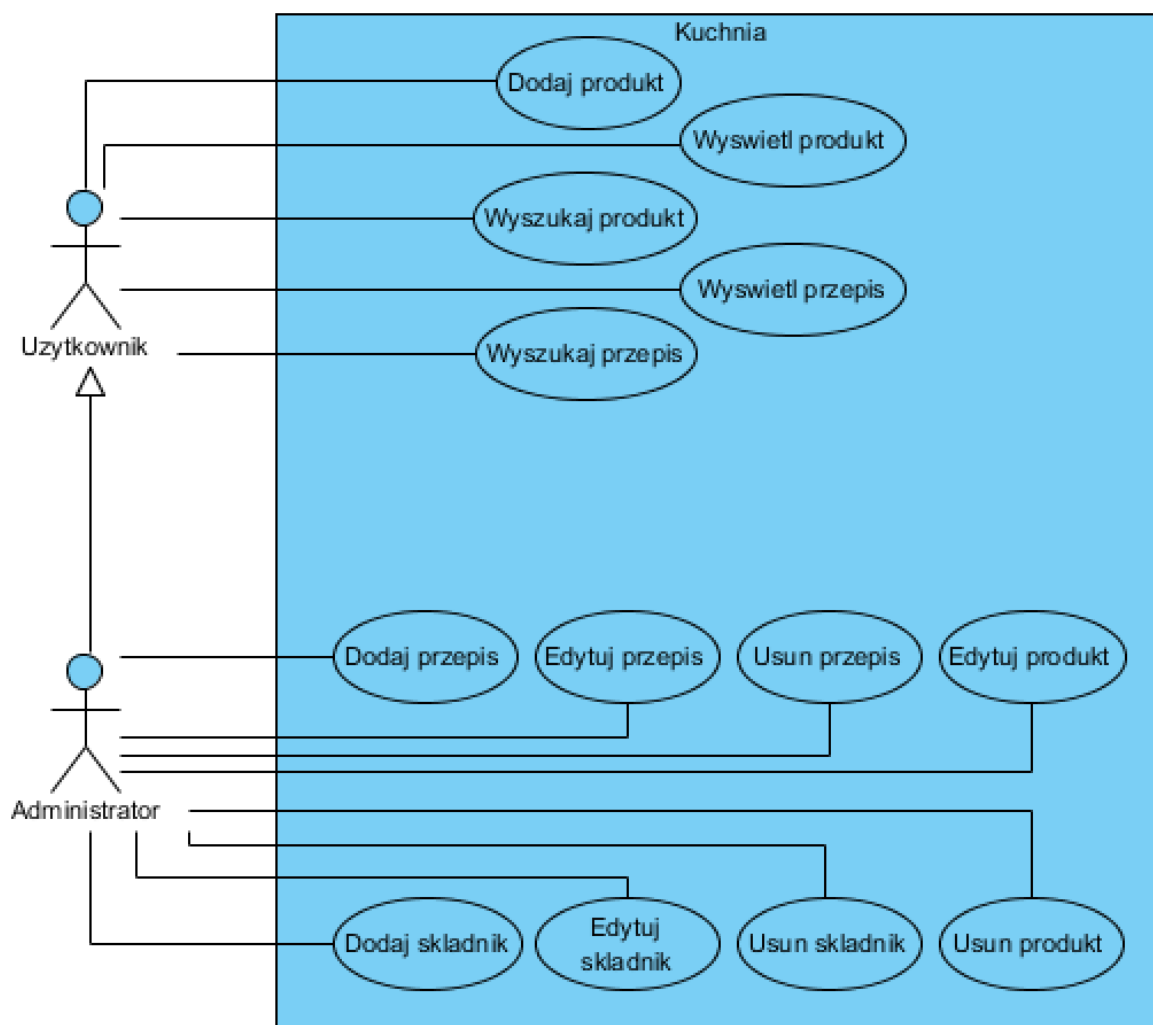
Rys.5 Przykładowe widoki

### 3.1.5. Projekt mechanizmów bezpieczeństwa na poziomie bazy danych

Dostęp do bazy danych odbywa się za pomocą tylko jednego konta, posiadającego pełne uprawnienia. Z racji nie przechowywania żadnych wrażliwych danych nie przewiduje się dodatkowej ochrony systemu na poziomie bazy danych.

## 3.2. Projekt aplikacji użytkownika

### 3.2.1. Architektura aplikacji i diagramy projektowe



Rys.6 Diagram przypadków użycia.



### 3.2.2. Interfejs graficzny i struktura menu

Interfejs graficzny zaprojektowany zostanie za pomocą HTML5 oraz CSS3.

START PRODUKTY PRZEPISY



Rys.7 Strona startowa.

START PRODUKTY PRZEPISY

WYŚWIETL WYSZUKAJ DODAJ USUN EDYTUJ

Dostępne produkty:

jajka  
maka ryzowa  
olej ryzowy  
skrobia ziemniaczana  
sol  
woda

Rys.8 Podstrona „Produkty”.

## Przepisy:

### nalesniki ryzowe

Rys.9 Podstrona „Przepisy”.

### 3.2.4. Metoda podłączania do bazy danych – integracja z bazą danych

W ramach posiadanego przeze mnie serwera wirtualnego w home.pl, dostęp do bazy danych MySQL można uzyskać poprzez skrypty PHP, znajdujące się na tym samym serwerze. Do połączenia z bazą postanowiłam użyć modułu mysqli, który umożliwia używanie bazy danych za pomocą składni PHP.

Dostęp do bazy danych odbywa się przez panel administracyjny (phpmyadmin), gdzie można tworzyć, modyfikować oraz usuwać tabele oraz dane umieszczone w bazie danych.

Tabela	Działanie	Rekordy	Typ	Metoda porównywania napisów	Rozmiar	Nadmiar
Produkty	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	6	InnoDB	utf8_polish_ci	80 KB	-
Przepisy	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	1	InnoDB	utf8_polish_ci	80 KB	-
Przepisy_Skladniki	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	6	InnoDB	utf8_polish_ci	32 KB	-
Skladniki	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	6	InnoDB	utf8_polish_ci	64 KB	-
W_PRODUKTY	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	6	Widok	---	-	-
W_PRZEPISY	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	1	Widok	---	-	-
W_SKLADNIKI	Przeglądaj   Struktura   Szukaj   Wstaw   Opróżnij   Usuń	6	Widok	---	-	-
<b>7 tabel</b>	<b>Suma</b>	<b>32</b>		<b>utf8_polish_ci</b>	<b>256 KB</b>	<b>0 B</b>

Rys.10 Panel administracyjny (phpmyadmin).

### 3.2.5. Projekt zabezpieczeń na poziomie aplikacji

Dostęp do bazy danych z poziomu aplikacji odbywa się za pomocą jednego konta, posiadającego pełne uprawnienia.

Do zabezpieczenia danych na stronie postanowiłam wykorzystać właściwości pliku .htaccess.

## 4. Implementacja systemu baz danych

### 4.1. Tworzenie tabel i definiowanie ograniczeń

Tabele powstały na podstawie poniższego kodu wygenerowanego w programie Visual Paradigm z diagramu ERD (z podpunktu 3.1.3). W kodzie zamieszczone są m.in. klucze główne oraz indeksy.

```
CREATE TABLE Produkty (  
  ID int(1000) NOT NULL AUTO_INCREMENT,  
  nazwa varchar(30) NOT NULL,  
  bezglutenowe tinyint,  
  lokalizacja varchar(20) NOT NULL,  
  szczegoly varchar(255),  
  ilosc float(10) NOT NULL,  
  jednostka_miary varchar(5) NOT NULL,  
  PRIMARY KEY (ID),  
  INDEX (ID),  
  INDEX (nazwa),  
  INDEX (bezglutenowe),  
  INDEX (lokalizacja));
```

```
CREATE TABLE Przepisy (  
  ID int(500) NOT NULL AUTO_INCREMENT,  
  nazwa varchar(30) NOT NULL,  
  bezglutenowe tinyint,  
  kategoria varchar(20),  
  opis text NOT NULL,  
  czas_przygotowania varchar(30),  
  SkladnikiID int(1000) NOT NULL,  
  PRIMARY KEY (ID),  
  INDEX (ID),  
  UNIQUE INDEX (nazwa),  
  INDEX (bezglutenowe),  
  INDEX (kategoria));
```

```
CREATE TABLE Przepisy_Skladniki (  
  PrzepisyID int(500) NOT NULL,  
  SkladnikiID int(5000) NOT NULL,  
  ID_PS int(15000) NOT NULL,  
  ilosc_skladnika float(10) NOT NULL,  
  jednostka_miary int(10) NOT NULL,  
  PRIMARY KEY (PrzepisyID, SkladnikiID, ID_PS));
```

```
CREATE TABLE Skladniki (  
  ID int(5000) NOT NULL AUTO_INCREMENT,  
  nazwa varchar(30) NOT NULL,  
  PRIMARY KEY (ID),  
  INDEX (ID),  
  UNIQUE INDEX (nazwa));
```

```
ALTER TABLE Przepisy_Skladniki ADD CONSTRAINT FKPrzepisy_S923814 FOREIGN KEY (PrzepisyID) REFERENCES Przepisy (ID);
```

```
ALTER TABLE Przepisy_Skladniki ADD CONSTRAINT FKPrzepisy_S716135 FOREIGN KEY (SkladnikiID) REFERENCES Skladniki (ID);
```

```
ALTER TABLE Produkty ADD CONSTRAINT FKProdukty286377 FOREIGN KEY (SkladnikiID) REFERENCES Skladniki (ID);
```

## 4.2. Implementacja mechanizmów przetwarzania danych

Skrypt tworzący widok W\_PRZEPISY:

```
CREATE ALGORITHM=MERGE VIEW 'W_PRZEPISY' AS SELECT * FROM 'Przepisy'
```

Skrypt tworzący widok W\_PRODUKTY:

```
CREATE ALGORITHM=MERGE VIEW 'W_PRODUKTY' AS SELECT * FROM 'Produkty'
```

Lista wszystkich widoków:

- W\_PRZEPISY
- W\_PRODUKTY

## 4.3. Implementacja uprawnień i innych zabezpieczeń

	Użytkownik	Administrator
Produkty	CRUD	CRUD
Składniki	R	CRUD
Przepisy	R	CRUD

Tabela 1 Uprawnienia

CRUD – Create (INSERT), Read (SELECT), Update (UPDATE), Delete (DELETE).

Użytkownik może zmieniać dane w ramach dostępnych produktów (których liczba będzie się zmieniać). Jednak w przypadku tabeli Składniki oraz Przepisy, użytkownik będzie mógł tylko wyświetlać ich zawartość. W celach bezpieczeństwa danych, tylko administrator może manipulować danymi obu tych tabel.

## 4.4. Testowanie bazy danych na przykładowych danych

Zapytanie numer 1:

Połączenie tabel Przepisy, Skladniki oraz Przepisy\_Skladniki:

```
SELECT P.nazwa, S.nazwa, PS.ilosc_skladnika, PS.jednostka_miary
```

```
FROM PRZEPISY AS P, Przepisy_Skladniki AS PS, Skladniki AS S
```

```
WHERE P.ID = PS.PrzepisyID
```

```
AND S.ID = SkladnikiID
```

```
ORDER BY P.ID
```

Wynik:

nazwa	nazwa	ilosc_skladnika	jednostka_miary
nalesniki ryzowe	skrobia ziemniaczana	2	łyżki
nalesniki ryzowe	maka ryzowa	120	g
nalesniki ryzowe	olej ryzowy	4	łyżki
nalesniki ryzowe	sol	0.25	łyżeczki
nalesniki ryzowe	jajka	3	szt
nalesniki ryzowe	woda	240	ml

Rys.11 Wynik zapytania numer 1.

Zapytanie numer 2:

Dodanie nowego produktu:

```
INSERT INTO `Produkty`(`ID`, `nazwa`, `bezglutenowe`, `lokalizacja`, `ilosc`, `jednostka_miary`)  
VALUES ('','test3','tak','szafka_2','1.5','ml')
```

ID	nazwa	bezglutenowe	lokalizacja	szczegoly	ilosc	jednostka_miary
1	woda	NULL	NULL	NULL	NULL	NULL
2	jajka	NULL	lodowka	NULL	10	szt
3	sol	NULL	szafka_2	NULL	1	kg
4	olej ryzowy	NULL	szafka_1	NULL	300	ml
5	maka ryzowa	tak	szafka_1	NULL	NULL	NULL
6	skrobia ziemniaczana	tak	szafka_1	NULL	300	g
8	test1	nie	szafka_1		2	kg
9	test2	nie	szafka_1		5	kg
10	test3	tak	szafka_2	NULL	2	ml

Rys.12 Tabela **Produkty** z dodanym produktem „test3”.

Zapytanie numer 3:

Modyfikacja nazwy z „test3” na „testt3”:

```
UPDATE `Produkty` SET `nazwa`='testt3'WHERE nazwa='test3';
```

ID	nazwa	bezglutenowe	lokalizacja	szczegoly	ilosc	jednostka_miary
1	woda	NULL	NULL	NULL	NULL	NULL
2	jajka	NULL	lodowka	NULL	10	szt
3	sol	NULL	szafka_2	NULL	1	kg
4	olej ryzowy	NULL	szafka_1	NULL	300	ml
5	maka ryzowa	tak	szafka_1	NULL	NULL	NULL
6	skrobia ziemniaczana	tak	szafka_1	NULL	300	g
8	test1	nie	szafka_1		2	kg
9	test2	nie	szafka_1		5	kg
10	testt3	tak	szafka_2	NULL	2	ml

Rys.13 Tabela **Produkty** po modyfikacji nazwy produktu.

Zapytanie numer 5:

```
INSERT INTO `28977644_friendly`.`Składniki` (`ID`, `nazwa`, `ProduktyID`)
```

```
VALUES (NULL, 'tescik', '6');
```

ID	nazwa	ProduktyID
2	woda	NULL
3	jajka	NULL
4	sol	NULL
5	olej ryzowy	NULL
6	maka ryzowa	NULL
7	skrobia ziemniaczana	NULL
8	tescik	6

Rys.15 Tabela **Składniki** po dodaniu nowego składnika i przypisaniu mu id produktu.

Zapytanie numer 4:

```
UPDATE `Przepisy` SET `bezglutenowe` = 'tak' WHERE `Przepisy`.`ID` =3;
```

ID	nazwa	bezglutenowe	kategoria	opis	czas_przygotowania
1	nalesniki ryzowe	tak	nalesniki	Mąki wymieszaj i przesiej. Jajka z solą ubij mikse...	30 min
3	test1	tak	ciasta		30 min

Rys.14 Tabela Przepisy po modyfikacji wartości zmiennej „bezglutenowe”.

Zapytanie numer 6:

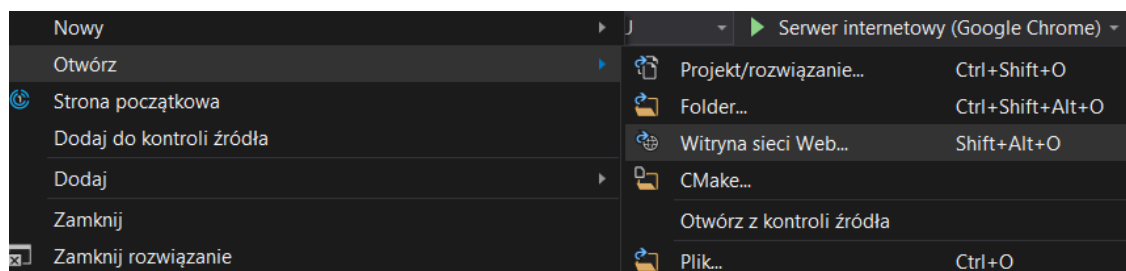
```
DELETE FROM `Produkty` WHERE nazwa='test2';
```

ID	nazwa	bezglutenowe	lokalizacja	szczegoly	ilosc	jednostka_miary
1	woda	NULL	NULL	NULL	NULL	NULL
2	jajka	NULL	lodowka	NULL	10	szt
3	sol	NULL	szafka_2	NULL	1	kg
4	olej ryzowy	NULL	szafka_1	NULL	300	ml
5	maka ryzowa	tak	szafka_1	NULL	NULL	NULL
6	skrobia ziemniaczana	tak	szafka_1	NULL	300	g
10	testt3	tak	szafka_2	NULL	2	ml

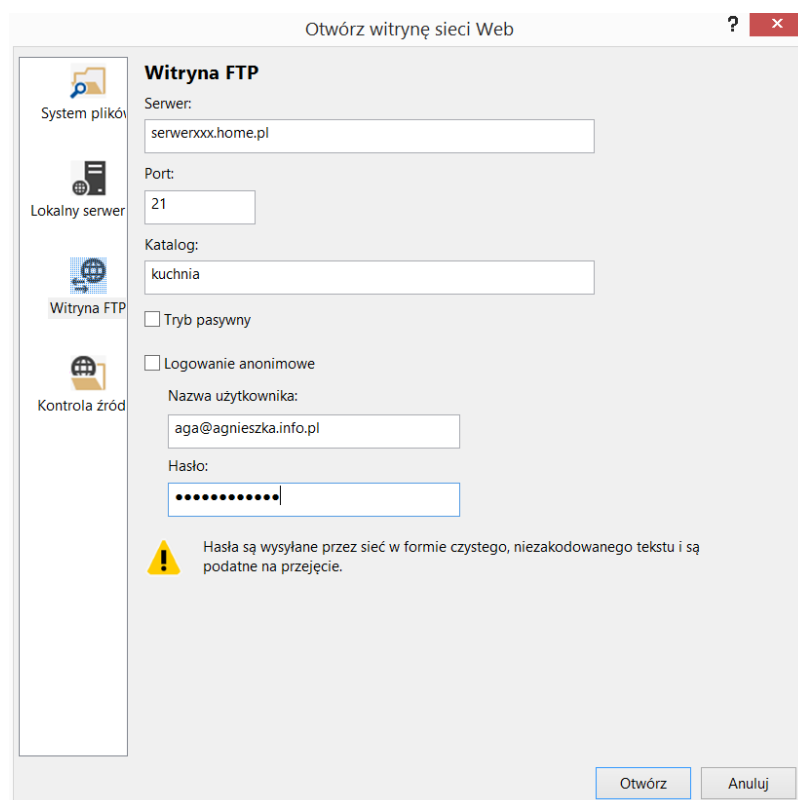
Rys.16 Tabela Produkty po usunięciu produktu o nazwie „test2”.

## 5. Implementacja i testy aplikacji

### 5.1. Instalacja i konfigurowanie systemu



Za pomocą środowiska Microsoft Visual Studio, połączyłam się za pomocą FTP z moim serwerem:



Dzięki temu, pliki były na bieżąco aktualizowane.

Do korzystania ze strony potrzebny jest dostęp do internetu oraz przeglądarka WWW.

### 5.2. Instrukcja użytkowania aplikacji

Aplikacja udostępnia przyjazny dla użytkownika interfejs ułatwiający korzystanie z dostępnych funkcjonalności systemu. Zrzuty ekranu zawarte w dalszej części dokumentacji prezentują działanie strony WWW.



### 5.3. Testowanie opracowanych funkcji systemu

The screenshot shows a web application interface with a navigation bar at the top containing the links: START, PRODUKTY, PRZEPISY, WYŚWIETL, WYSZUKAJ, and DODAJ. Below the navigation bar, there is a search section. It starts with the text 'Wyszukaj według:' followed by a dropdown menu currently showing 'Nazwa'. Below this, there is a text input field labeled 'Szukane wyrażenie:' and a button labeled 'Prześlij'.

Rys.17 Próba wyszukania produktu, bez wprowadzenia parametrów wyszukiwania.

The screenshot shows the same web application interface as in Rys.17. Below the search section, there is a message: 'Brak parametrów wyszukiwania, wróć do poprzedniej strony i spóbj ponownie!'.

Rys.18 Informacja o nie wprowadzeniu parametrów wyszukiwania.

START   PRODUKTY   PRZEPISY  
WYŚWIETL   WYSZUKAJ

Wyszukaj według:

Szukane wyrażenie:

Rys.19 Wyszukanie przepisu według kategorii - wprowadzenie parametrów wyszukiwania.

START   PRODUKTY   PRZEPISY  
WYŚWIETL   WYSZUKAJ

Nazwa: naleśniki ryzowe  
Czas przygotowania: 30 min  
Bezglutenowe: tak  
Kategoria: naleśniki

Opis: Mąki wymieszaj i przesiej. Jajka z solą ubij mikserem na pianę (1-2 minuty). Powoli dodawaj mąkę, a następnie, ciągle miksując, wlewaj powoli wodę. Na końcu dodaj olej i jeszcze chwilę miksuj. Odstaw ciasto na kwadrans. Patelnię rozgrzej i wlej na nią łyżkę oleju. Zamieszaj ciasto i porcję ciasta wlej na patelnię, najlepiej, by naleśniki były cienutkie. Smaż na średnim ogniu. Po około minucie przełóż na drugą stronę. Nie smaż zbyt długo, bo naleśniki staną się sztywne i trudne do nadziewania farszem.

Rys.20 Wyszukanie przepisu według kategorii - wyniki wyszukiwania.

## 5.4. Omówienie wybranych rozwiązań programistycznych

### 5.4.1. Implementacja interfejsu dostępu do bazy danych

Aby połączyć się z bazą danych stworzyłam plik connect.php, którego zawartość wygląda następująco:

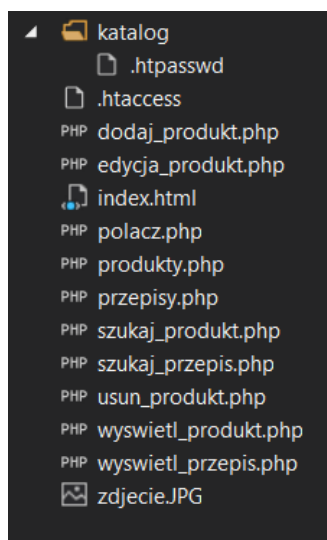
```
<?php
    $servername = "serwerxxx.home.pl";
    $username = "xxx";
    $password = "xxx";
    $dbname = "xxx";
?>
```

Następnie wystarczy wstawić poniższy skrypt do każdego fragmentu kodu, gdzie chcemy się odwołać do bazy danych:

```
<?php
include_once 'connect.php';
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$conn->close();
?>
```

#### 5.4.2. Implementacja wybranych funkcjonalności systemu



Rys.21 Struktura plików.

Wyszukanie produktu:

```
<div class="stylefont2">
  <form action="wyniki_produkt.php" method="post">
    Wyszukaj według:
    <select name="metoda">
      <option value="nazwa" />Nazwa
      <option value="lokalizacja" />Lokalizacja
    </select>
    <br /><br />
    Szukane wyrażenie:
    <input type="text" name="wyrazenie" />
    <input type="submit" name="wyszukaj" />
  </form>
</div>
```

Rys.22 Szukanie produktu - HTML.

```

<?php
$metoda = $_POST['metoda'];
$wyrazenie = $_POST['wyrazenie'];
$wyrazenie = trim($wyrazenie);
if (!$metoda || !$wyrazenie)
{
    echo 'Brak parametrów wyszukiwania, wróć do poprzedniej strony i spróbuj ponownie!';
    exit;
}
if (!get_magic_quotes_gpc())
{
    $metoda = addslashes($metoda);
    $wyrazenie = addslashes($wyrazenie);
}
// połączenie z bazą danych w osobnym pliku
include_once 'polacz.php';
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$conn->query('SET NAMES utf8');
$conn->query('SET CHARACTER SET utf8_unicode_ci');
$zapytanie = "select * from W_PRZEPISY where ".$metoda. " like '%" . $wyrazenie . "%'";
$wynik = $conn->query($zapytanie);
$ile_znaleziono = $wynik->num_rows;
for ($i=0;$i<$ile_znaleziono;$i++)
{
    $wiersz = $wynik->fetch_assoc();
    echo 'Nazwa: ' . stripslashes($wiersz['nazwa']) . '<br />';
    echo 'Opis: ' . stripslashes($wiersz['opis']) . '<br />';
    echo 'Bezglutenowe: ' . stripslashes($wiersz['bezglutenowe']) . '<br />';
    echo 'Składniki: ' . stripslashes($wiersz['ilosc']) . '<br />';
    echo ' ' . stripslashes($wiersz['jednostka_miary']) . '<br /><br /><br />';
}
$wynik->free();
$conn->close();
?>

```

Rys.23 Szukanie produktu - PHP.

```

<?php
include_once 'polacz.php';
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if(isset($_POST['save']))
{
    $sql = "INSERT INTO W_PRODUKTY (nazwa, bezglutenowe, lokalizacja, szczegoly, ilosc, jednostka_miary)
VALUES ('".$_POST["nazwa"]."', '".$_POST["bezglutenowe"]."', '".$_POST["lokalizacja"]."', '".$_POST["szczegoly"]."', '".$_POST["ilosc"]."', '".$_POST["jednostka_miary"]."')";
    $result = mysqli_query($conn,$sql);
}
$conn->close();
?>

<div class="stylefont2">

<form action="produkty.php" method="post">
<label id="first">Nazwa: </label><br />
<input type="text" name="nazwa" required<br />
<label id="first"> Bezglutenowe (tak/nie): </label><br />
<input type="text" name="bezglutenowe"><br />
<label id="first"> Lokalizacja: </label><br />
<input type="text" name="lokalizacja"><br />
<label id="first"> Szczegóły: </label><br />
<input type="text" name="szczegoly"><br />
<label id="first"> Ilość: </label><br />
<input type="float" name="ilosc"><br />
<label id="first"> Jednostka miary (g, ml): </label><br />
<input type="text" name="jednostka_miary"><br />

<button type="submit" name="save">Dodaj</button>

</form>
</div>

```

Rys.24 Dodanie produktu.

```

<div class="stylefont2">
Dostępne produkty:<br /><br />
<?php
// połączenie z bazą danych w osobnym pliku
include_once 'polacz.php';
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM W_PRODUKTY";
$result = $conn->query($sql);

if ($result->num_rows > 0)
{
    // output data of each row
    while($row = $result->fetch_assoc())
    {
        echo " " . $row["nazwa"]. " " . $row["bezglutenowe"]. " " . $row["lokalizacja"]. " " . $row["szczegoly"]. " " . $row["ilosc"]. " " . $row["jednostka_miary"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
</div>

```

Rys.25 Fragment kodu odpowiedzialny za wyświetlenie produktów.

```

<?php
// połączenie z bazą danych w osobnym pliku
include_once 'polacz.php';
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT nazwa FROM W_PRZEPISY";
$result = $conn->query($sql);

if ($result->num_rows > 0)
{
    // output data of each row
    while($row = $result->fetch_assoc())
    {
        echo " " . $row["nazwa"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

Rys.26 Fragment kodu odpowiedzialny za wyświetlenie przepisów.

### 5.4.3. Implementacja mechanizmów bezpieczeństwa

Zawartość pliku .htaccess:

```

AuthName "Podaj hasło"
AuthType Basic
AuthUserFile /web/serwerxxx.home.pl/kuchnia/katalog/.htpasswd
Require valid-user

```

Zawartość pliku .htpasswd w katalogu „Katalog” składa się z loginu oraz hasła, zakodowanego algorytmem crypt():

agnieszka:DfAnGzZo/2yoM

## 6. Podsumowanie i wnioski

Przedłożony projekt jest odpowiedzią na zauważony problem z utrzymaniem porządku w kuchni, a w rozwiązaniu tego problemu pomóc ma uporządkowanie kuchennej zawartości w bazie danych. Wychodzi się bowiem z założenia, że jeśli można sprawnie sprawdzić lokalizację tego, co się szuka, to z łatwością można to znaleźć. Przy czym trzeba zwrócić uwagę na to, że bezproblemowe korzystanie z takiej bazy danych wymaga aktualizowania listy produktów, które znajdują się na stanie – tylko wtedy bowiem, od razu widać po pierwsze to, czego brakuje, po drugie to, co się kończy, po trzecie to, co trzeba skończyć, żeby się nie zmarnowało. Zauważa się bowiem, że zapominanie o tym, co już zostało kupione nierzadko kończy się wyrzucaniem żywności z powodu jej przeterminowania, a takie sytuacje są szczególnie niepożądane nie tylko z punktu widzenia jednostkowego gospodarstwa domowego, lecz ponadto globalnie patrząc na problematykę z tym związaną.

W realizacji projektu częstym utrudnieniem był fakt, że w ramach serwera, na którym została postawiona baza danych, nie było dostępu do triggerów, a dostęp do plików .htaccess jest utrudniony. Strona internetowa została napisana przy wykorzystaniu HTML, CSS oraz PHP. Niestety moja mała znajomość ostatniego języka, spowodowała, że nie wszystkie funkcje działają jak należy.

## Literatura

Kurs PHP. Dostęp[27.12.2018] Dostępny w Internecie: <https://phpkurs.pl/podstawy/>

W3schools. Dostęp[27.12.2018] Dostępny w Internecie: <https://www.w3schools.com/sql/>