



**WYŻSZA SZKOŁA  
INFORMATYKI i ZARZĄDZANIA**  
z siedzibą w Rzeszowie

## **KOLEGIUM INFORMATYKI STOSOWANEJ**

**Kierunek: INFORMATYKA**

**Specjalność: Programowanie**

Michał Biesiadecki

Jaromir Stawiarski

Wojciech Lesiczka

Nr albumu studentów: w67220, w67269, w67249

***Aplikacja webowa „ByteCuisine”***

Prowadzący: mgr inż. Ewa Żesławska

**Praca projektowa programowanie obiektowe C#**

**Rzeszów 2024**



# Spis treści

<b>Wstęp</b>	<b>4</b>
<b>1 Opis założeń projektu</b>	<b>5</b>
1.1 Cele projektu . . . . .	5
1.2 Wymagania funkcjonalne i niefunkcjonalne . . . . .	6
<b>2 Opis struktury projektu</b>	<b>7</b>
2.1 Klient . . . . .	7
2.2 Serwer . . . . .	8
2.3 Shared . . . . .	9
2.4 Baza danych . . . . .	10
<b>3 Harmonogram realizacji projektu</b>	<b>12</b>
<b>4 Prezentacja warstwy użytkowej projektu</b>	<b>13</b>
4.1 Login page . . . . .	13
4.2 Home page . . . . .	14
4.3 Nawigacja . . . . .	15
4.4 Fridge page . . . . .	16
4.4.1 Zarządzanie wirtualną lodówką . . . . .	17
4.4.2 Wyszukiwanie składników . . . . .	17
4.4.3 Wyszukiwanie przepisów . . . . .	18
4.4.4 Lista brakujących składników . . . . .	18
4.5 Settings page . . . . .	19
4.6 Admin page . . . . .	20
4.7 Statistics page . . . . .	21
<b>5 Podsumowanie</b>	<b>22</b>
<b>Bibliografia</b>	<b>23</b>
<b>Spis rysункów</b>	<b>24</b>

# **Wstęp**

Wyobraź sobie, że стоisz przed swoją lodówką, drzwi są uchylone, zastanawiasz się nad wiekowym dilematem, co ugotować. W środku znajduje się mieszanka losowych składników, które nie sugerują od razu jasnej opcji posiłku. Masz dość tych samych przepisów i nie jesteś pewien, jak kreatywnie połączyć te składniki. Wyzwanie polega nie tylko na przygotowaniu posiłku, ale na stworzeniu czegoś satysfakcjonującego i ekscytuującego z tego, co masz w lodówce. W dobie cyfrowej transformacji, nasz projekt kieruje się ku innowacyjnemu podejściu w kulinarnym świecie cyfrowym. Zamiarem tego projektu jest stworzenie platformy cyfrowej, która przerodzi tradycyjne przeglądanie przepisów kulinarnych w pełne, spersonalizowane doświadczenie kulinarne. Naszym celem jest uczynienie planowania posiłków procesem intuicyjnym, przyjemnym i dostosowanym do indywidualnych potrzeb użytkownika.

# Rozdział 1

## Opis założeń projektu

### 1.1 Cele projektu

Stworzenie innowacyjnej platformy cyfrowej, która przekształci tradycyjne przeglądanie przepisów kulinarnych w spersonalizowane doświadczenie kulinarne, uczynienie procesu planowania posiłków intuicyjnym, przyjemnym i dostosowanym do indywidualnych potrzeb użytkowników.

- **Problem do rozwiązania i jego źródło:** Głównym problemem jest brak kreatywności i personalizacji w tradycyjnych metodach planowania posiłków. Użytkownicy często napotykają trudności w tworzeniu posiłków z ograniczonych lub losowych składników dostępnych w domu, co prowadzi do rutyn kulinarnych i zmniejszonej satysfakcji z gotowania.
- **Znaczenie problemu i dowody jego istnienia:** Problem ten jest istotny, ponieważ wpływa na codzienne doświadczenia kulinarne wielu osób. Rutynowe podejście do gotowania może prowadzić do nudnych posiłków i zmniejszenia zainteresowania gotowaniem. Zainteresowanie personalizowanymi rozwiązaniami kulinarnymi jest widoczne w trendach rynkowych oraz w rosnącej popularności aplikacji i narzędzi do planowania posiłków.
- **Wymagania do rozwiązania problemu:** Do rozwiązania tego problemu niezbędna jest głęboka analiza potrzeb użytkowników, opracowanie zaawansowanych algorytmów dopasowujących przepisy do dostępnych składników i preferencji użytkowników, oraz stworzenie intuicyjnego interfejsu użytkownika, który ułatwi proces planowania posiłków.
- **Metoda rozwiązania problemu i rezultaty:** Projekt rozpocznie się od badań rynkowych i ankietyzacji potencjalnych użytkowników w celu zrozumienia ich potrzeb. Następnie opracowany zostanie prototyp platformy, który będzie zawierał algorytmy bazujące na sztucznej inteligencji do personalizacji przepisów i planowania posiłków. Projekt zakończy się wdrożeniem i testowaniem platformy, która ma być łatwa w użyciu i dostępna na różne urządzenia. Wynikiem końcowym będzie w pełni funkcjonalna aplikacja mobilna i/lub strona internetowa, która oferuje spersonalizowane doświadczenie kulinarne.

## **1.2 Wymagania funkcjonalne i niefunkcjonalne**

### **Wymagania funkcjonalne**

- Logowanie - walidacja danych podczas logowania
- Zmiana hasła użytkownika - walidacja danych podczas zmiany hasła
- Przeglądanie przepisów - interfejs umożliwiający przeglądanie dostępnych przepisów, wyszukiwanie przepisów według składników umieszczonych w wirtualnej lodówce
- Wybieranie składników - funkcja umożliwiająca użytkownikowi składników do wirtualnej lodówki ze zbioru składników, sugerowanie przepisów na podstawie wybranych składników
- Dodawanie przepisów, składników oraz użytkowników - możliwość dodawania nowych rekordów do bazy danych poprzez importowanie plików CSV z danymi
- Współpraca z bazą danych - skonfigurowanie bazy danych do przechowywania przepisów, informacji o użytkownikach
- Możliwość wyszukiwania składników oraz dań w listach
- Dynamiczne aktualizacje treści dodaniu składnika do wirtualnej lodówki

### **Wymagania niefunkcjonalne**

- Czas ładowania strony - strona powinna ładować się w ciągu maksymalnie 5 sekund
- Zgodność z przeglądarkami - strona powinna być zgodna z przynajmniej dwiema przeglądarkami
- Responsywność interfejsu użytkownika - interfejs powinien być dostosowany w ciągu 2 sekund od zmiany rozmiaru ekranu
- Czas odpowiedzi bazy danych na zapytania - czas odpowiedzi nie powinien przekraczać 1 sekundy dla większości zapytań

# Rozdział 2

## Opis struktury projektu

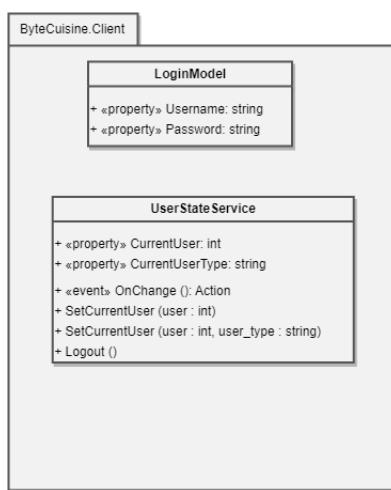
W projekcie wykorzystywane są następujące języki programowania i projektowania: C#, Javascript, HTML oraz CSS. Połączenie tych technologii zapewnia nam framework Blazor, który bazuje na języku C#. W projekcie użyto wersji aplikacja zestawu WebAssembly platformy Blazor, która charakteryzuje się podziałem na 3 struktury przez co łatwiej można wpływać na zasoby wykorzystywane po stronie klienta i serwera. Dodatkowo występuje również część shared, która jest wspólna dla klienta i serwera. Do połączenia się z bazą danych wykorzystaliśmy EntityFramework, który umożliwia komunikację między aplikacją i bazą, a także zapewnia możliwość korzystania z migracji.

### 2.1 Klient

Część Client jest to kod wykonywany po stronie użytkownika. W Client są dwie klasy LoginModel oraz UserStateService.

**Klasa LoginModel** - znajduje się LoginPage.razor i służy do porównywania danych logowania z bazy z tymi, które są wprowadzane przez użytkownika podczas logowania.

**Klasa UserStateService** - przechowuje identyfikator oraz rolę zalogowanego użytkownika oraz udostępnia te informacje do innych klas w projekcie. Te informacje wykorzystywane są do poprawnego wyświetlania pozycji w nawigacji aplikacji oraz poprawnego przypisywania składników do lodówki zalogowanego użytkownika.



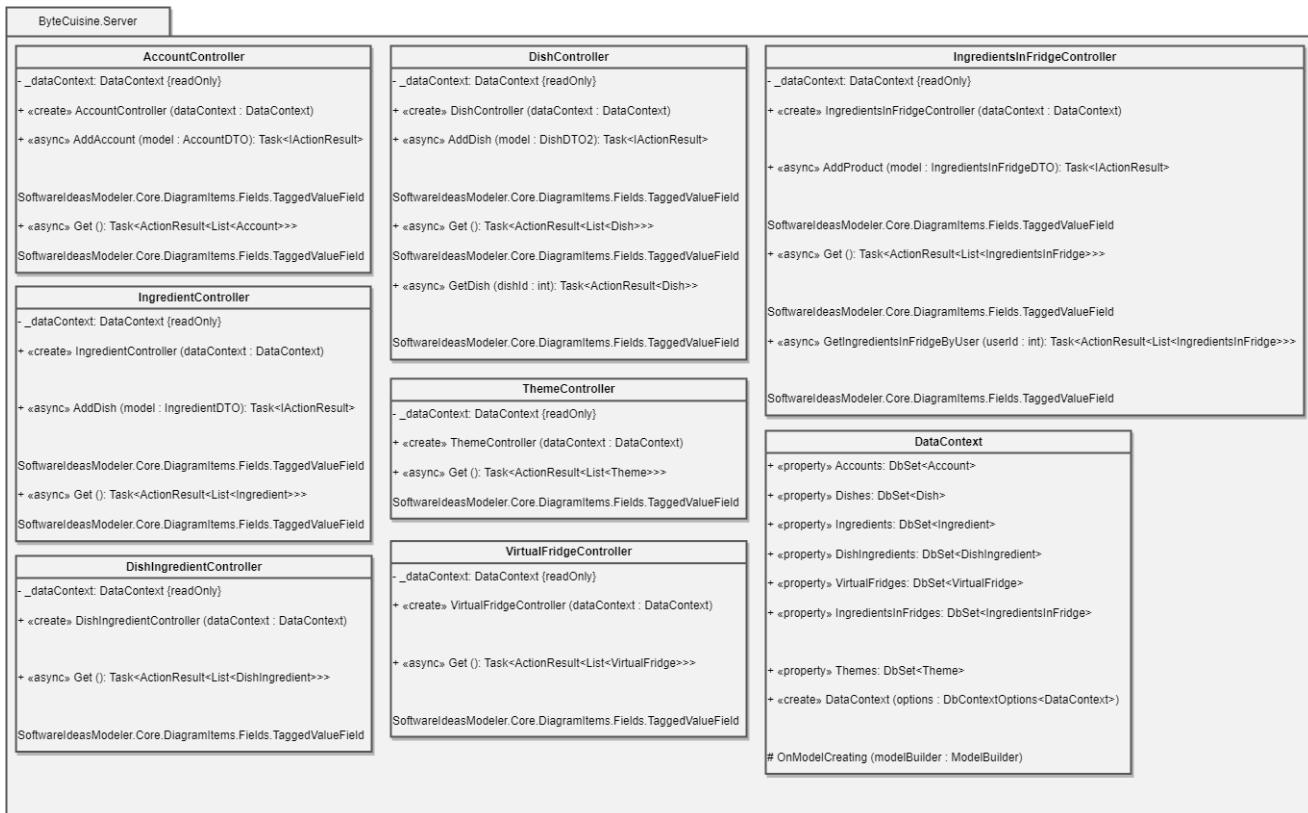
Rysunek 2.1: Diagram klas po stronie klienta

## 2.2 Serwer

Część serwerowa jest to kod wykonywany po stronie dostawcy aplikacji. W serwerze jest wiele klas, które pełnią funkcję kontrolerów. W Blazor WebAssembly, kontrolery mogą być wykorzystywane w zapleczu (backend), zazwyczaj w osobnym projekcie, do obsługi zapytań HTTP, takich jak operacje CRUD (Create, Read, Update, Delete) na danych. Kontrolery obsługują punkty końcowe API, które służą jako ścieżki, którymi przesyłane są informacje między klientem, a serwerem. Aplikacja Blazor WebAssembly komunikuje się z backendem za pomocą żądań HTTP, często wykorzystując HttpClient. Oznacza to, że wysyła żądania do kontrolerów, które następnie przetwarzają te żądania, komunikując się z bazą danych lub innymi usługami, a następnie zwracają odpowiedź w postaci JSON. Ważnym elementem części serwerowej są kontrolery HTTP, które obsługują żądania z klienta. Poniżej przedstawiono przykładowe implementacje kontrolerów dla żądań typu `HttpGet` i `HttpPost`. Kontroler `HttpGet` służy do pobierania danych z serwera. Poniżej znajduje się przykład kontrolera, który zwraca listę wszystkich użytkowników.

Kontroler `HttpPost` jest używany do przesyłania danych na serwer. Poniżej znajduje się przykład kontrolera, który umożliwia dodanie nowego użytkownika.

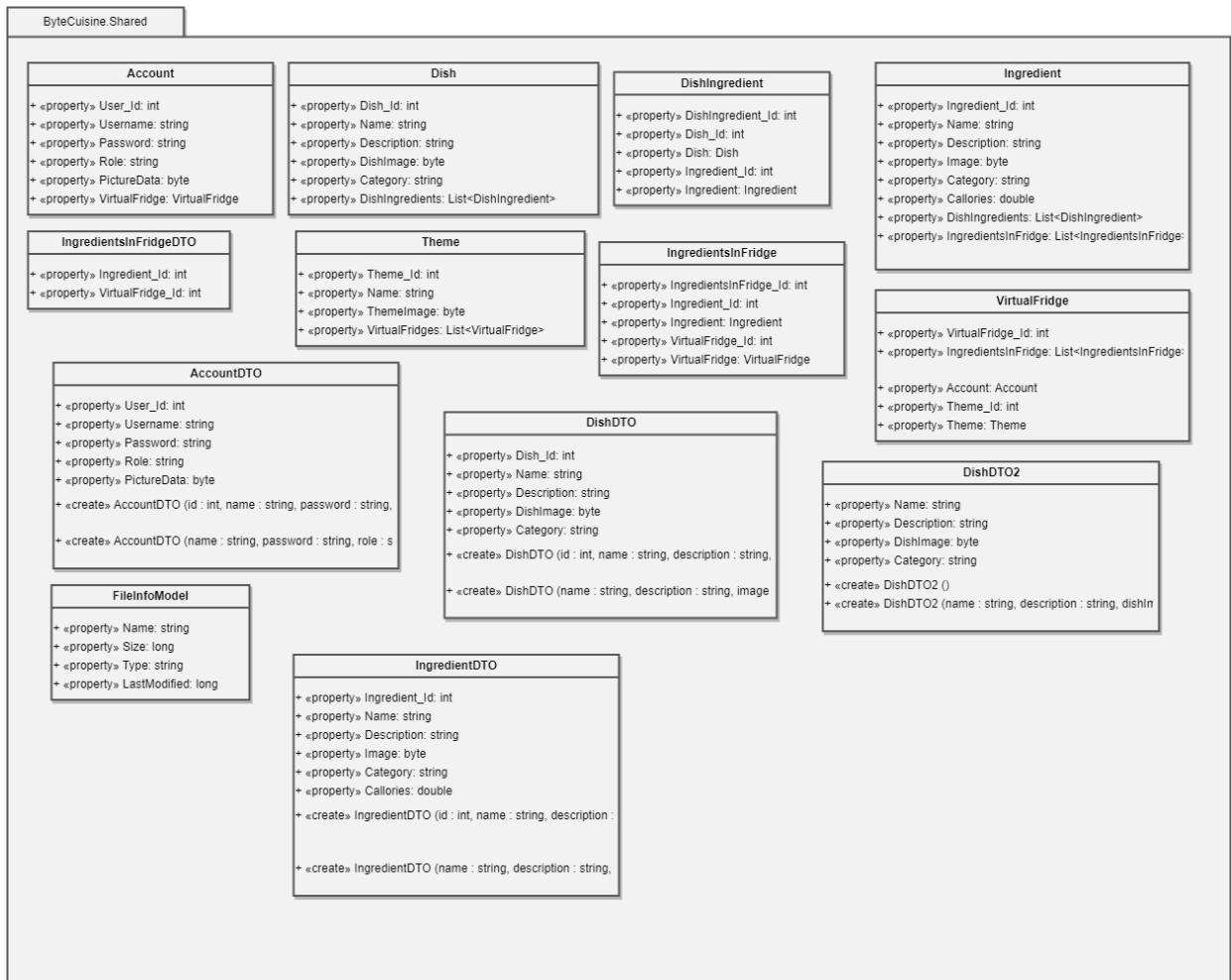
W serwerze znajduje się plik `appsettings.json`, w którym zawarty jest connection string. Connection string zapewnia nam połączenie z bazą danych poprzez ustawienie nazwy serwera (bazy danych) i nazwy bazy danych w wyżej wymienionym pliku.



Rysunek 2.2: Diagram klas po stronie serwera

## 2.3 Shared

Shared jest to część wspólna aplikacji, która polega na dzieleniu kodu pomiędzy serwer, a klienta. Część wspólna służy głównie do zapisu i odczytu z bazy danych, a także do importu i eksportu plików CSV. Ponadto w projekcie występują klasy typu DTO, które mają swoje zastosowanie z kontrolerach za pomocą których odbywają się operacje na bazie danych.



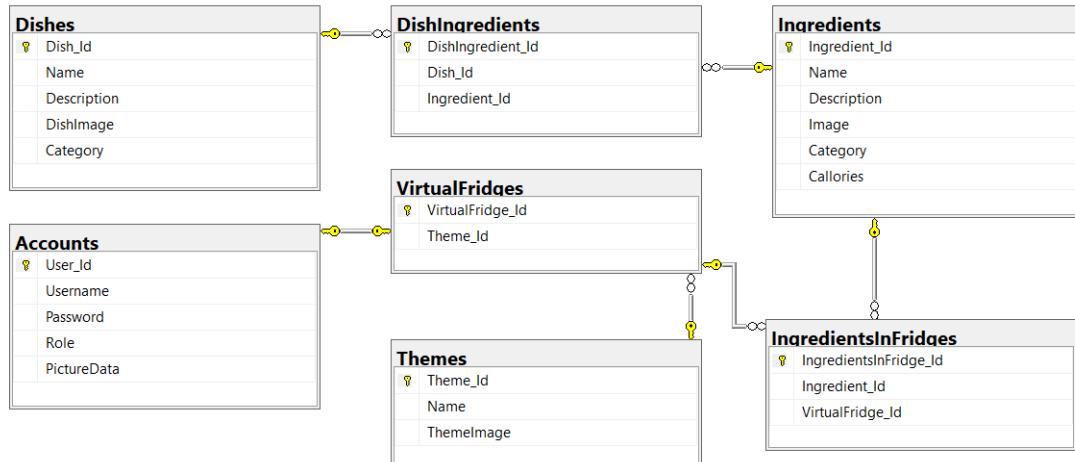
Rysunek 2.3: Diagram klas dla wspólnego środowiska

## 2.4 Baza danych

Baza danych powstała poprzez wykorzystanie 7 klas podczas tworzenia migracji za pomocą EntityFramework. Zbiór informacji zabiera następujące szczegóły:

- Tabela "Accounts":
  - Tabela przechowuje informacje o użytkownikach.
  - Kolumny:
    - \* "User\_Id": Identyfikator użytkownika (klucz główny).
    - \* "Username": Nazwa użytkownika.
    - \* "Password": Hasło użytkownika.
    - \* "Role": Rola użytkownika (np. "Admin" lub "User").
    - \* "PictureData": Dane obrazu użytkownika w formacie varbinary.
- Tabela "Dishes":
  - Tabela zawiera informacje o potrawach.
  - Kolumny:
    - \* "Dish\_Id": Identyfikator potrawy (klucz główny).
    - \* "Name": Nazwa potrawy.
    - \* "Description": Opis potrawy.
    - \* "DishImage": Dane obrazu potrawy w formacie varbinary.
    - \* "Category": Kategoria potrawy (np. "Italian Food" lub "Asian Food").
- Tabela "DishIngredients":
  - Tabela przechowuje informacje o składnikach potraw.
  - Kolumny:
    - \* "DishIngredient\_Id": Identyfikator relacji składnik-potrawa (klucz główny).
    - \* "Dish\_Id": Identyfikator potrawy (klucz obcy).
    - \* "Ingredient\_Id": Identyfikator składnika (klucz obcy).
- Tabela "Ingredients":
  - Tabela zawiera informacje o składnikach.
  - Kolumny:
    - \* "Ingredient\_Id": Identyfikator składnika (klucz główny).
    - \* "Name": Nazwa składnika.
    - \* "Description": Opis składnika.
    - \* "Image": Dane obrazu składnika w formacie varbinary.
    - \* "Category": Kategoria składnika (np. "Italian" lub "Vegetable").
    - \* "Calories": Kalorie składnika (liczba zmiennoprzecinkowa).

- Tabela "IngredientsInFridges":
  - Tabela przechowuje informacje o składnikach w lodówkach wirtualnych.
  - Kolumny:
    - \* "IngredientsInFridge\_Id": Identyfikator relacji składnik-lodówka wirtualna (klucz główny).
    - \* "Ingredient\_Id": Identyfikator składnika (klucz obcy).
    - \* "VirtualFridge\_Id": Identyfikator lodówki wirtualnej (klucz obcy).
- Tabela "Themes":
  - Tabela zawiera informacje o motywach (np. kolorach).
  - Kolumny:
    - \* "Theme\_Id": Identyfikator motywu (klucz główny).
    - \* "Name": Nazwa motywu.
    - \* "ThemeImage": Dane obrazu motywu w formacie varbinary.
- Tabela "VirtualFridges":
  - Tabela przechowuje informacje o lodówkach wirtualnych.
  - Kolumny:
    - \* "VirtualFridge\_Id": Identyfikator lodówki wirtualnej (klucz główny).
    - \* "Theme\_Id": Identyfikator motywu przypisanego do lodówki wirtualnej (klucz obcy).

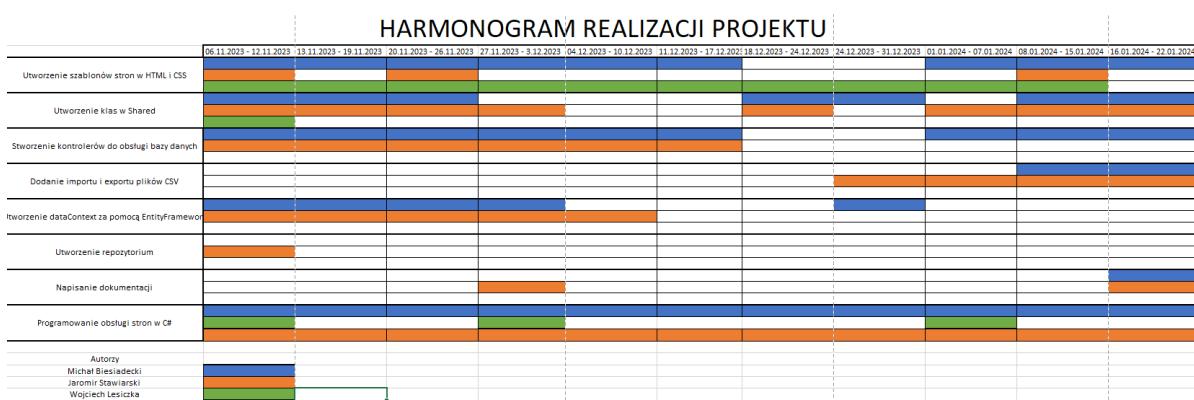


Rysunek 2.4: Diagram bazy danych

## Rozdział 3

# Harmonogram realizacji projektu

Harmonogram pracy składał się z kilku składowych, które są widoczne na rysunku 3.1. Prace nad projektem zaczęły się 12 listopada poprzez utworzenie repozytorium (znajduje się pod adresem <https://github.com/Terrnox/ByteCuisine>) tak, aby każdy z członków zespołu mógł mieć dostęp do aktualnej wersji projektu i mógł wrzucać kod do repozytorium. Wykorzystywanym systemem kontroli wersji jest Git, który jest zaimplementowany jako rozszerzenie do Visual Studio. Jednym kluczowych elementów było utworzenie szablonu w HTML, ponieważ bez tego nie byłoby możliwe wyświetlanie się danych. Równocześnie w tym samym czasie ruszyły prace nad tworzeniem kontrolerów, które miały obsługiwać operacje CRUD wykonywane w aplikacji. Jednak podczas ich kreacji napotkaliśmy na problem, który uniemożliwił zastosowanie wcześniej utworzonych klas (wykorzystane do przechowywania danych z bazy) do wstrzykiwania danych do bazy. Ta przeszkoda została rozwiązana poprzez zastosowanie klas pomocniczych z końcówką DTO (Data Transfer Object). Kolejne problemy pojawiły się podczas dodawania importu i eksportu plików CSV do projektu. W celu dodania danych znajdujących się poza projektem potrzebny był skrypt napisany w Javascript. Jednak podczas importowania owszych plików natrafiliśmy na zjawisko wyścigu, które występuje w asynchronicznych funkcjach wykorzystywanych w naszej aplikacji. Tą zagwostkę rozwiązaliśmy poprzez dodanie dwóch dodatkowych elementów referencyjnych, które służyły jako oddzielne punkty referencyjne dla plików. Prace zakończyły się po napisaniu dokumentacji.



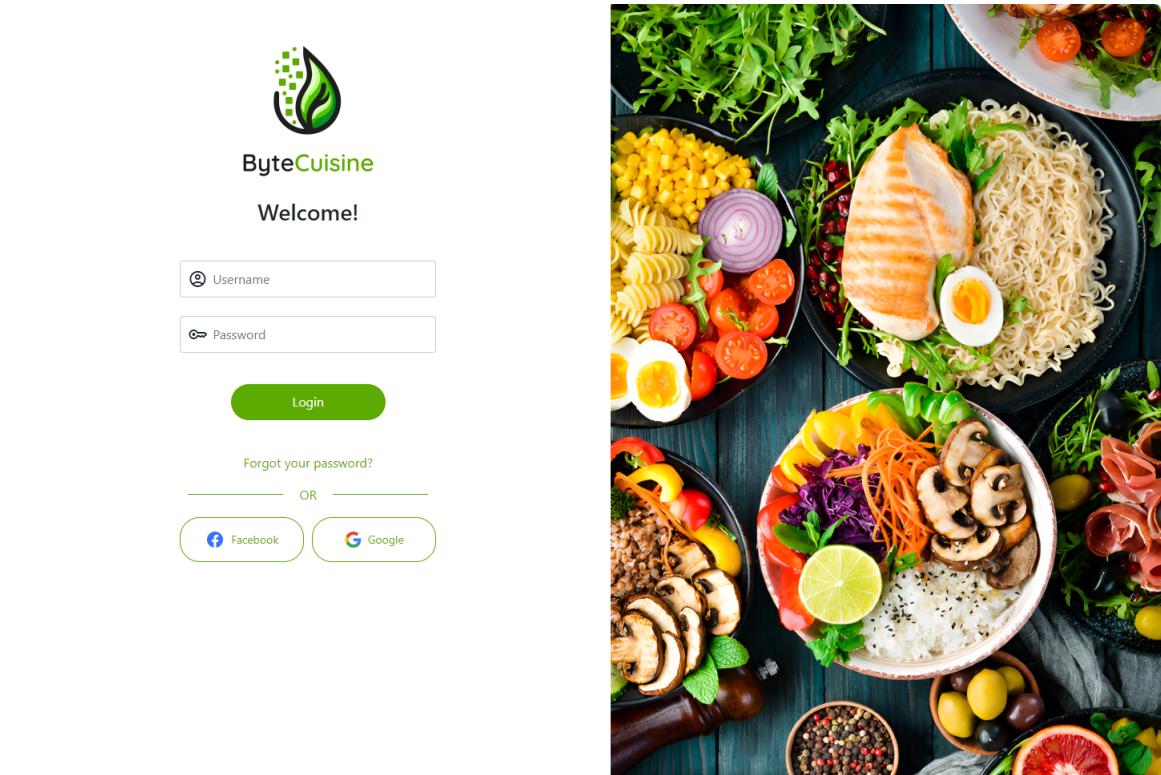
Rysunek 3.1: Harmonogram pracy nad projektem

## Rozdział 4

# Prezentacja warstwy użytkowej projektu

### 4.1 Login page

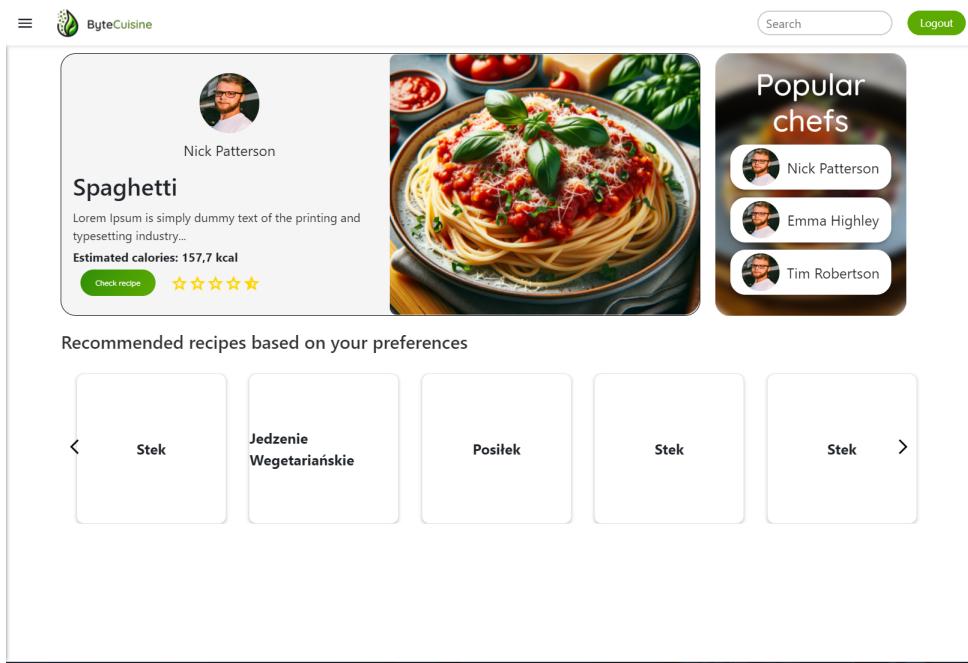
Użytkownik uruchamiający aplikację zostaje przywitany poprzez stronę logowania, która posiada dwa input boxy, do których użytkownik przekazuje dane. Strona ukazująca się użytkownikowi po zalogowaniu, zależy od roli, która jest przypisana do jego konta. W przypadku roli 'User' pojawi się strona Home, natomiast w przypadku, gdy użytkownik posiada rolę 'Admin' pojawi się panel admina.



Rysunek 4.1: Strona logowania do aplikacji

## 4.2 Home page

Na stronie Home znajduje się podstawowa implementacja strony głównej, która posiada górny panel przedstawiający najlepszy przepis dnia wraz z rankingiem najlepiej prosperujących twórców przepisów danego miesiąca. W dolnej części strony użytkownik, może znaleźć karuzelę, która będzie przedstawała proponowane dla niego przepisy na podstawie danych gromadzonych podczas codziennego korzystania z aplikacji. Wyżej wymienione funkcjonalności zaimplementowane będą w późniejszych etapach projektowania, ponieważ nie są one uwzględnione w wymaganiach projektowych oraz ze względu na dużą złożoność proponowanych rozwiązań.



Rysunek 4.2: Strona główna dla usera

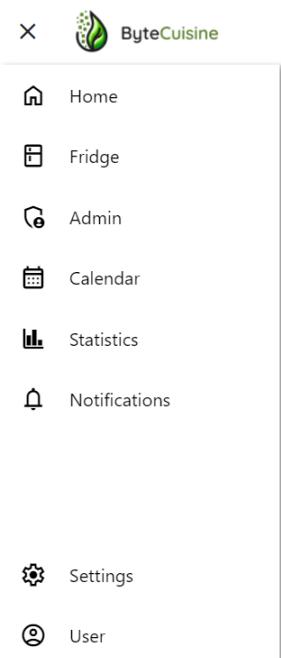
## 4.3 Nawigacja

Nawigacja stanowi kluczowy element interfejsu użytkownika naszej aplikacji, zapewnia dostęp do różnych funkcji oraz sekcji. Wspólny element stanowi górny pasek - 4.3, posiadający przycisk do otwierania oraz zamykania bocznego menu, pasek wyszukiwania (na razie brak funkcjonalności) oraz zielony przycisk do wylogowania użytkownika z konta.

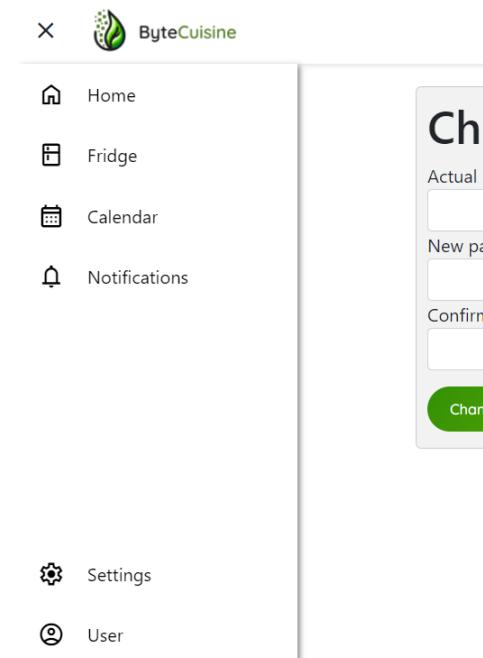


Rysunek 4.3: Navbar

W zależności od tego, jaką rolę posiada zalogowany użytkownik boczne menu może różnić się swoją zawartością. Dwa stany bocznej nawigacji przedstawiają zdjęcia - 4.4 oraz 4.5.



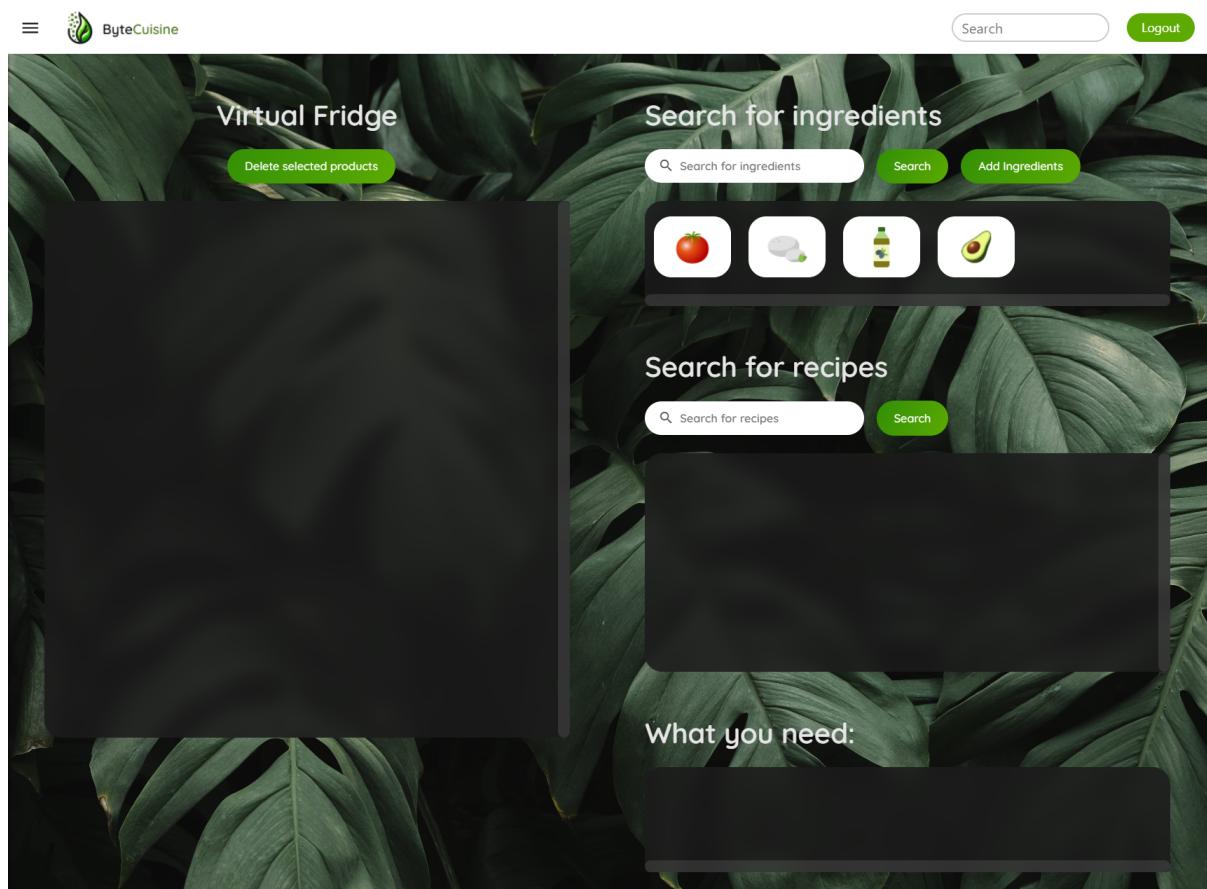
Rysunek 4.4: Side menu dla admina



Rysunek 4.5: Side menu dla usera

## 4.4 Fridge page

Strona wirtualnej lodówki jest głównym elementem naszej aplikacji i oferuje użytkownikom wiele przydatnych funkcjonalności związanych z zarządzaniem produktami spożywczymi, przepisami oraz planowaniem posiłków. Pomaga w znalezieniu przepisów na potrawy na podstawie składników dostępnych w wirtualnej lodówce. To miejsce, gdzie użytkownicy mogą kontrolować swoje zapasy, znaleźć inspirację kulinarną i korzystać z przydatnych narzędzi do zarządzania kuchnią.

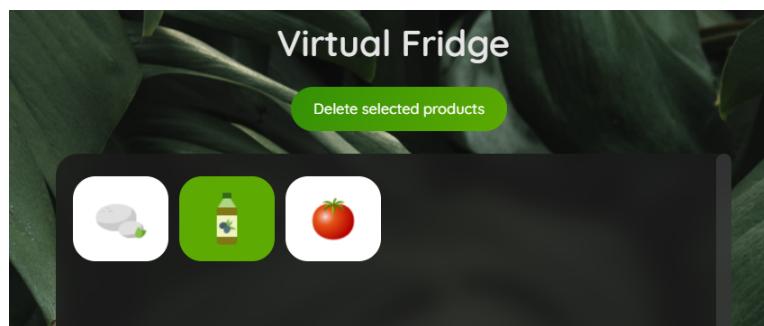


Rysunek 4.6: Strona wirtualnej lodówki

Poniżej przedstawiam opis głównych funkcji tej strony:

#### 4.4.1 Zarządzanie wirtualną lodówką

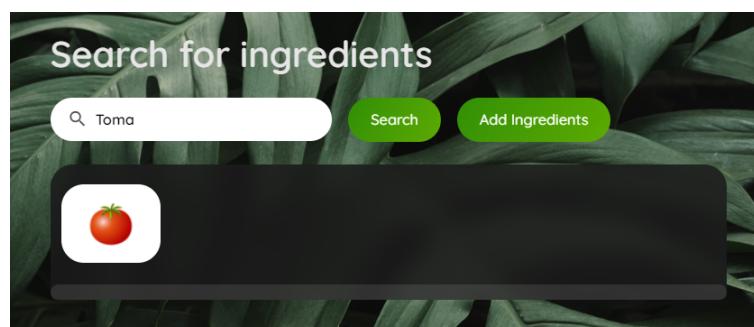
- Użytkownicy mogą zobaczyć listę produktów w swojej wirtualnej lodówce w lewym panelu.
- Każdy produkt jest reprezentowany za pomocą miniatury obrazka pobranego z bazy danych (lub komunikatu "No image available", jeśli brak obrazu).
- Możliwość zaznaczania produktów do usunięcia, a następnie usunięcia ich za pomocą przycisku "Delete selected products".
- Wygodne wybieranie produktów do usunięcia poprzez kliknięcie na nie.



Rysunek 4.7: Zarządzanie wirtualną lodówką

#### 4.4.2 Wyszukiwanie składników

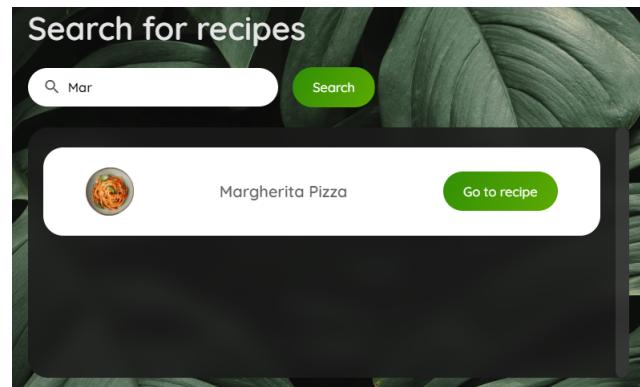
- Użytkownicy mogą wyszukiwać składniki spożywcze w bazie danych.
- Wyniki wyszukiwania są wyświetlane w formie przefiltrowanej listy produktów z miniaturami obrazków.
- Możliwość zaznaczania wybranych składników do dodania do wirtualnej lodówki, przy jednoczesnym zaznaczeniu produktu



Rysunek 4.8: Wyszukiwanie składnika

#### 4.4.3 Wyszukiwanie przepisów

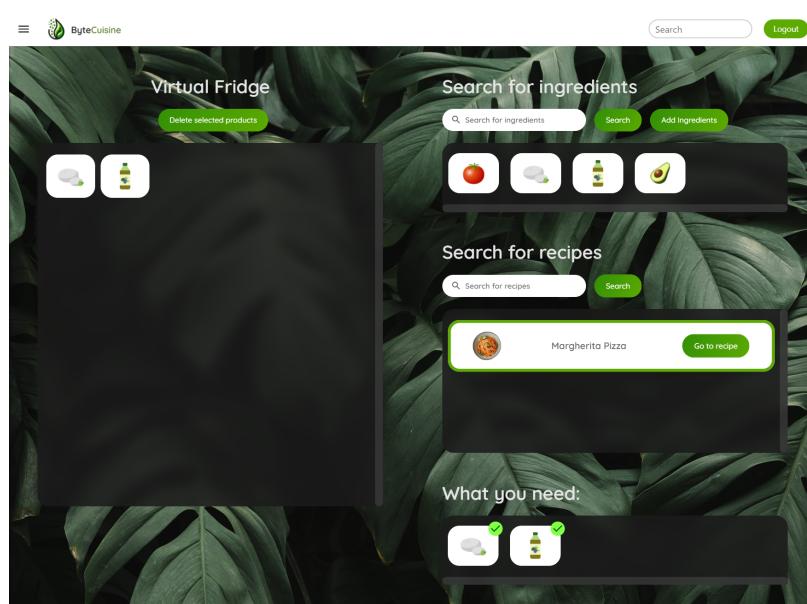
- Użytkownicy mogą wyszukiwać przepisy kulinarnie.
- Wyniki wyszukiwania są wyświetlane w formie przefiltrowanej listy potraw z miniaturami obrazków.
- Możliwość zaznaczania wybranych przepisów do wyświetlenia ich szczegółów.
- Automatyczne filtrowanie wyników wyszukiwania na bieżąco w zależności od wprowadzonego zapytania.



Rysunek 4.9: Wyszukiwanie przepisu

#### 4.4.4 Lista brakujących składników

- Dla wybranych przepisów użytkownicy mogą sprawdzić, jakich składników im brakuje.
- Brakujące składniki są wyświetlane w formie przefiltrowanej listy, z oznaczeniem jako posiadane lub brakujące.
- Produkty posiadane oraz brakujące w wirtualnej lodówce są automatycznie rozpoznawane.



Rysunek 4.10: Lista brakujących składników

## 4.5 Settings page

Każdy z użytkowników posiada możliwość zmiany hasła poprzez przejście do zakładki Settings (wybierane na pobocznym menu). Po przejściu do tej strony osoba ujrzy formularz, który dotyczy zmiany hasła. W celu zmiany hasła należy podać aktualne hasło (Actual password), nowe hasło (New password) oraz powtórzyć jeszcze raz nowe hasło (Confirm a new password). Procedurę należy zakończyć klikając w przycisk 'Change password'.

The screenshot shows the 'Change a password' form. It includes fields for 'Actual password', 'New password', and 'Confirm a new password'. Below the fields is a green 'Change password' button. At the bottom of the form, there is a green horizontal bar with the text 'Wrong password!'. The top navigation bar features the ByteCuisine logo, a search bar, and a 'Logout' button.

Rysunek 4.11: Zmiana hasła

W przypadku wpisania niepoprawnego hasła użytkownik zobaczy komunikat o nieprawidłowej procedurze zmiany hasła.

The screenshot shows the 'Change a password' form with an error message. The 'Actual password' field contains three dots ('...'), the 'New password' field contains four dots ('....'), and the 'Confirm a new password' field also contains four dots ('....'). Below these fields is a green 'Change password' button. At the bottom of the form, there is a green horizontal bar with the text 'Wrong password!'. The top navigation bar features the ByteCuisine logo, a search bar, and a 'Logout' button.

Rysunek 4.12: Niedana zmiana hasła

W przeciwnym wypadku, gdy wprowadzi się poprawne dane hasło zostanie zmienione i zapisane do bazy danych.

The screenshot shows the 'Change a password' form with a success message. The 'Actual password' field contains three dots ('...'), the 'New password' field contains four dots ('....'), and the 'Confirm a new password' field also contains four dots ('....'). Below these fields is a green 'Change password' button. At the bottom of the form, there is a green horizontal bar with the text 'Password changed successfully!'. The top navigation bar features the ByteCuisine logo, a search bar, and a 'Logout' button.

Rysunek 4.13: Udana zmiana hasła

## 4.6 Admin page

Po zalogowaniu na konto admina oczom użytkownika pojawi się strona poświęcona adminowi, na której może zobaczyć informacje znajdujące się w bazie danych. Administrator będzie mógł zobaczyć dane 5 rekordów naraz, a także całkowitą liczbę wierszy jednej tabeli. Jeśli ilość rzędów danych jest większa niż 5 wtedy można skorzystać z przycisku do przechodzenia do następnych stron, w celu ich wyświetlenia.

The screenshot shows a web application titled "Data In Database" with three separate sections for different tables:

- User Table:** Shows 2 items. The first row has User\_Id 1, Username "admin", Role "Admin", and an image placeholder. The second row has User\_Id 2, Username "tom", Role "User", and an image placeholder.
- Dish Table:** Shows 2 items. The first row has Dish\_Id 1, Name "Margherita Pizza", Category "Italian", Description "Classic Italian pizza with tomatoes, mozzarella cheese, fresh basil, salt, and olive oil.", and two image placeholders. The second row has Dish\_Id 2, Name "Sushi Roll", Category "Japanese", Description "Traditional Japanese sushi roll with fresh fish, rice, and seaweed.", and two image placeholders.
- Ingredient Table:** Shows 0 items.

Each section includes a "Export To CSV" button at the bottom.

Rysunek 4.14: Strona dla admina

Dodatkowo admin ma opcję wyświetlenia danych zawartych w plikach CSV pod warunkiem, że importowany plik będzie zawierał jedną z poniżej podanych struktur:

```
User_Id;Username;Password;Role;PictureData  
Dish_Id;Name;Description;DishImage;Category  
Ingredient_Id;Name;Description;Image;Category;Callories
```

Wszystkie zdjęcia muszą zawierać opis obrazka w formacie base64. Po kliknięciu w przycisk 'Wybierz plik' pojawi się okno z wyborem plików z formatem CSV.

The screenshot shows a web application titled "Data In CSV File" with three separate sections for different tables:

- User Table:** Shows 0 items. It has a "wybierz plik" button and a message "Nie wybrano pliku".
- Dish Table:** Shows 0 items. It has a "wybierz plik" button and a message "Nie wybrano pliku".
- Ingredient Table:** Shows 0 items. It has a "wybierz plik" button and a message "Nie wybrano pliku".

Each section includes an "Insert to database" button at the bottom.

Rysunek 4.15: Import danych

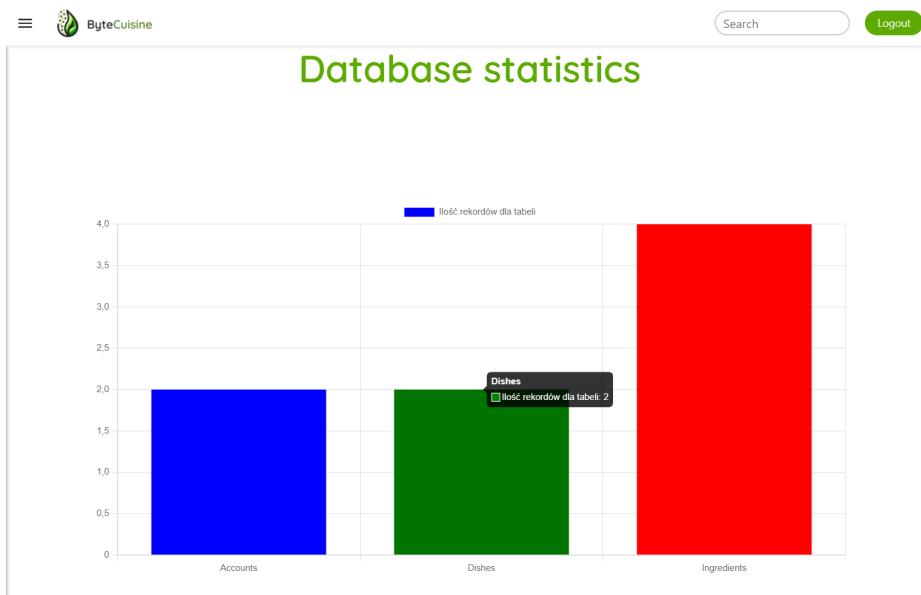
Po wybraniu odpowiedniego zbioru danych pojawi się informacja o nazwie wybranego pliku, a po chwili zostaną wyświetlane dane, które znajdują się w otwieranym pliku. Aplikacja posiada również funkcje dodania informacji do bazy danych poprzez naciśnięcie przycisku 'Insert To Database', lecz zimportowane zostaną tylko te dane, które nie znajdują się w bazie danych.

The screenshot shows a user interface titled 'Data In CSV File'. It displays four separate tables, each with its own header and data. The first table, 'User', has columns 'User\_Id', 'Username', 'Role', and 'Image', with two items listed. The second table, 'Dish', has columns 'Dish\_Id', 'Name', 'Category', 'Description', and 'Image', with zero items listed. The third table, 'Ingredient', has columns 'Ingredient\_Id', 'Name', 'Description', 'Category', 'Calories', and 'Image', with zero items listed. The fourth table is completely blank. Each table includes a green 'Insert to database' button at the bottom right.

Rysunek 4.16: Rezultat importu

## 4.7 Statistics page

Admin ma również opcję zobaczenia statystyk, które pokazują ilość danych w trzech głównych tabelach: Accounts, Dishes, Ingredients. W celu zobaczenia tych statystyk należy wybrać zakładkę Statistics. Dane są przedstawione w postaci wykresu słupkowego.



Rysunek 4.17: Statystyka

# Rozdział 5

## Podsumowanie

Podczas realizowania projektu zrealizowaliśmy podstawowe funkcjonalności opisane w założeniach projektowych z przedmiotu Programowanie Obiektowe C# - Projekt. W ramach realizacji projektu zaimplementowaliśmy następujące funkcjonalności:

- GUI użytkownika wykonane na podstawie HTML oraz kaskadowych arkuszy stylów
- import oraz eksport danych do plików CSV w AdminPage
- nawiązywanie połączenia z bazą danych, dzięki której aplikacja jest w stanie wykonywać operacje CRUD
- wirtualna lodówka, w której użytkownik może dodawać i usuwać składniki, a dzięki tym składnikom sugerowane są możliwe potrawy do wykonania.
- statystyki ilości rekordów w bazie danych
- kompatybilność z wieloma przeglądarkami takimi jak: Microsoft Edge, Google Chrome, Brave oraz Opera GX
- możliwość zmiany hasła zalogowanego użytkownika

W dalszych pracach rozwojowych nad projektem planujemy dodać następujące funkcjonalności:

- wyliczanie ilości kalorii dla przepisu na podstawie użytych składników
- zabezpieczenia aplikacji tak, aby w pełni korzystała z funkcji Authorization, w celu uniemożliwienia dostępu do danych wrażliwych
- funkcję rejestracji z wykorzystaniem własnego E-maila lub dostępu poprzez platformy Google oraz Facebook
- obsługę ocen potraw przez użytkowników
- opcję znalezienia najbliższego sklepu z danym składnikiem, który potrzebny jest do wykonania wybranej potrawy
- forum dla użytkowników aplikacji

# Bibliografia

- [1] <https://www.w3schools.com/cs/index.php> z dnia 21.01.2024 C Tutorial
- [2] <https://learn.microsoft.com/pl-pl/dotnet/architecture/blazor-for-web-forms-developers> z dnia 21.01.2024 Blazor dla deweloperów ASP.NET Web Forms
- [3] Jacek Matulewski, C#: lekcje programowania: praktyczna nauka programowania dla platform .NET i .NET Core, Helion, Gliwice 2021 lub nowsze.
- [4] R. S. Miles, C#: zacznij programować!, Helion, Gliwice 2020 lub nowsze.
- [5] Włodzimierz Gajda, Git : rozproszony system kontroli wersji, Helion, Gliwice 2013 lub nowsze.

# Spis rysunków

2.1	Diagram klas po stronie klienta . . . . .	7
2.2	Diagram klas po stronie serwera . . . . .	8
2.3	Diagram klas dla wspólnego środowiska . . . . .	9
2.4	Diagram bazy danych . . . . .	11
3.1	Harmonogram pracy nad projektem . . . . .	12
4.1	Strona logowania do aplikacji . . . . .	13
4.2	Strona główna dla usera . . . . .	14
4.3	Navbar . . . . .	15
4.4	Side menu dla admina . . . . .	15
4.5	Side menu dla usera . . . . .	15
4.6	Strona wirtualnej łodówki . . . . .	16
4.7	Zarządzanie wirtualną łodówką . . . . .	17
4.8	Wyszukiwanie składnika . . . . .	17
4.9	Wyszukiwanie przepisu . . . . .	18
4.10	Lista brakujących składników . . . . .	18
4.11	Zmiana hasła . . . . .	19
4.12	Nieuudana zmiana hasła . . . . .	19
4.13	Udana zmiana hasła . . . . .	19
4.14	Strona dla admina . . . . .	20
4.15	Import danych . . . . .	20
4.16	Rezultat importu . . . . .	21
4.17	Statystyka . . . . .	21