

DOKUMENTACJA

Książka Kucharska

Ewa Wydra

1. Temat

Tematem mojego projektu było napisanie aplikacji pełniącej rolę książki kucharskiej, korzystającej z operacji CRUDowych (Create Read Update Delete). Ponadto zawarta w niej jest opcja wyszukiwania przepisów według autora, nazwy a także użytych składników. Aplikacja także wyświetla QR code, który po odczytaniu wyświetlałby treść przepisu. Program został napisany w języku C#.

2. Analiza tematu

2.1. Doprecyzowanie tematu

Aplikacja przechowuje bazę przepisów za pomocą SQLite. Mieści się ona w pliku `CookingBook.sqlite`. Aby móc pracować bazie, aplikacja korzysta z narzędzia ORM – EntityFramework w wersji 6. Interfejs aplikacji został utworzony za pomocą WPF, dodatkowo GUI aplikacji zostało wzbogacone o zewnętrzny framework mahaps.

2.2. Uzasadnienie wyboru klas

Klasy modeli zostały wygenerowane automatycznie przez VisualStudio za pomocą podłączenia predefiniowanego pliku bazy danych stworzonego w technologii sqlite.

Na modele składają się następujące klasy:

Klasa `Recipe`, w której przechowywane są informacje o nazwie oraz treści przepisu. Zawiera ona także klucz obcy klasy `Author`, z którą jest w stosunku wiele do jednego, ponieważ jedna osoba może być autorem wielu przepisów. Kolejnym kluczem obcym jest `RecipeCategoryID`. Klasa `RecipeCategory` przechowuje informację o kategorii, do której zalicz się przepis. Do wyboru są: „Zupa”, „Śniadanie”, „Danie główne”, „Deser” oraz „Inne”. Jest ona powiązana z klasą `Recipe` w stosunku jeden do wielu.

Kolejną klasą powiązaną z klasą `Recipe` jest klasa `Ingredient`. Jest to klasa przechowująca informacje o składniku: jego nazwę oraz ilość. Jest ona powiązana z klasą `Recipe` przy pomocy klucza obcego `RecipeID` w stosunku wiele do jednego, ponieważ jeden przepis może zawierać wiele składników. Ponadto w klasie mieści się klucz obcy `TypeID`.

Klasa `QuantityType` przechowuje różne typy ilości: sztuki, łyżki, łyżeczki itd. Jest ona powiązana z klasą `Ingredient` w stosunku jeden do jednego.

Poza klasami służącymi do budowania bazy danych istnieją jeszcze klasy pełniące rolę View Modeli:

`AddRecipePageViewModel` – odpowiada za funkcjonalność strony służącej do dodawania oraz edycji przepisów.

`HomePageViewModel` – zajmuje się funkcjonalnością strony głównej, która otwiera się po uruchomieniu programu.

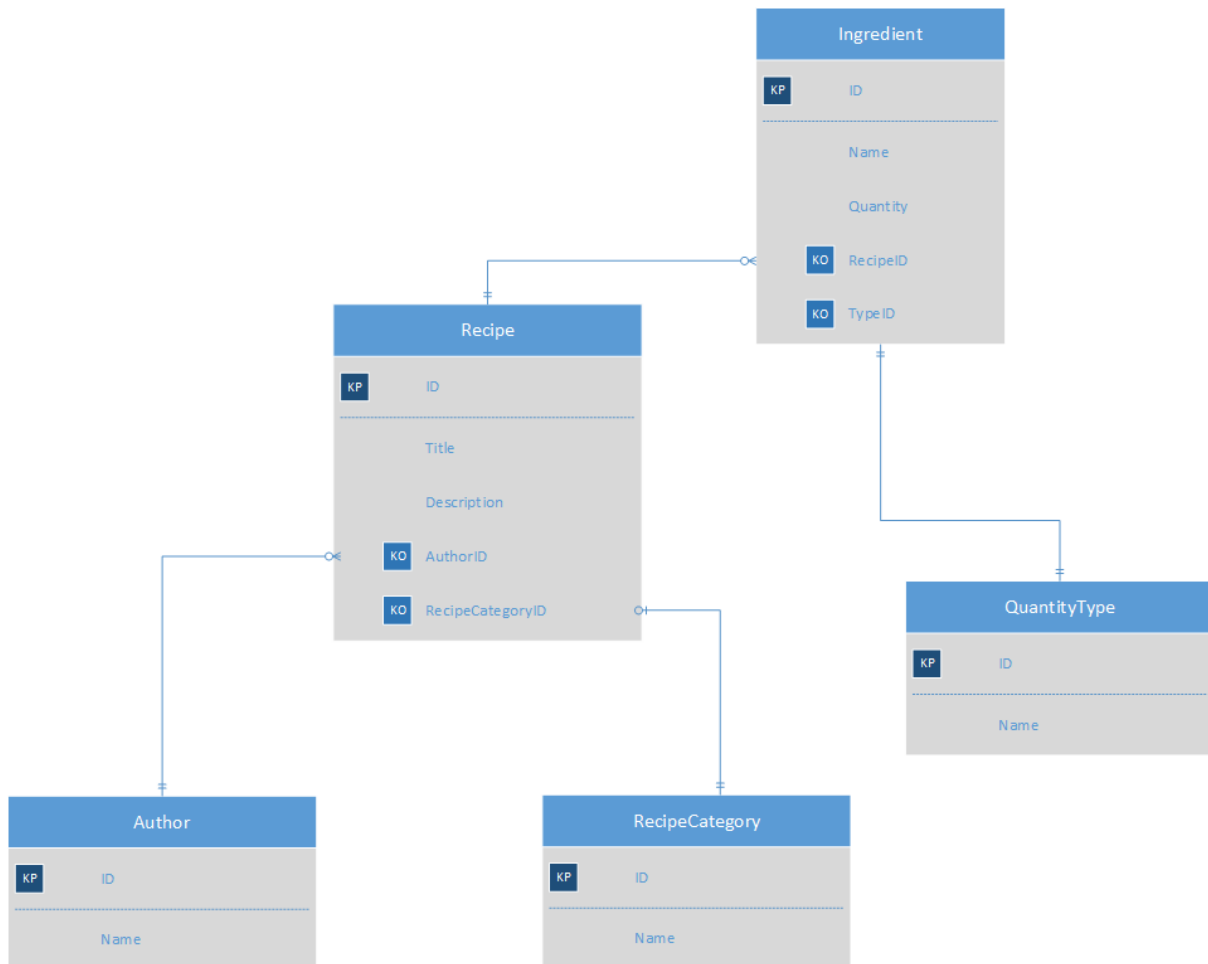
`MainViewModel` – odpowiada za główne okno oraz za działanie przycisków menu głównego.

RelayCommand – klasa skopiowana z MSDN Magazine, umożliwiająca zastosowanie wzorca architektonicznego MVVM

SearchPageViewModel – w niej zaimplementowana jest funkcjonalność strony wyszukiwania.

SharedVM – klasa statyczna służy do współdzielenia właściwości i funkcji publicznych między view modelami.

ShowRecipePageViewModel – odpowiada za funkcjonalność strony wyświetlającej przepis.



2.3 Biblioteki

W programie użyto kilka zewnętrznych bibliotek.

- EntityFramework – jest to biblioteka będąca narzędziem ORM, przeznaczona do budowania aplikacji opartych o bazy danych.
- System.Data.SQLite – biblioteka implementująca silnik SQLite.
- ZXing.NET – biblioteka służąca do tworzenia QR code.
- MahApps.Metro – biblioteka służąca do poprawienia GUI aplikacji.

3. Specyfikacja zewnętrzna

3.1. Widoki

3.1.1. Strona główna:

Na stronę główną jest uruchamiana przy starcie programu. Można również znaleźć się na tym widoku przez naciśnięcie przycisku „strona główna” znajdującego się w menu głównym po lewej stronie. Na górze strony głównej, pod nazwą aplikacji znajduje się menu. Naciśnięcie poszczególnych przycisków powoduje filtrowanie listy przepisów znajdującej się poniżej. Aby ponownie wyświetlić wszystkie przepisy, należy nacisnąć przycisk „Wszystkie”. Poniżej menu występuje prosta wyszukiwarka. Wyszukuje ona przepisy ze względu na ich nazwę. Wpisanie interesującej nas frazy do okienka tekstowego, a następnie naciśnięcie przycisku „Szukaj”, spowoduje wyświetlenie listy przepisów zawierających tę frazę. Po wybraniu interesującego nas przepisu poprzez naciśnięcie na jego pozycję w liście przepisów możemy skorzystać z trzech opcji: „Wybierz przepis” – przycisk który przeniesie nas do okna wyświetlania przepisu, „Edytuj przepis” – naciśnięcie przycisku spowoduje przełączenie okna na okno edycji przepisu, „Usuń przepis” – naciśnięcie przycisku spowoduje wyświetlenie komunikatu „Czy na pewno chcesz usunąć przepis?”. Wybranie opcji „Tak” spowoduje usunięcie wybranego przez nas przepisu z bazy danych.

KSIĄŻKA KUCHARSKA

szukaj

dodaj przepis

strona główna

wyjście

KSIĄŻKA KUCHARSKA

Zupy Desery Dania główne Śniadania Inne Wszystkie

SZUKAJ

Racuchy z jabłkami / Krysia
Ciasteczka pełnoziarniste / Ania
Krupnik / Krysia
Jaglanka z bananem i sosem z masła orzechowego / Ania
Grzany cydr / Andrzej
Kokosowe musli z truskawkami / Janusz

WYBIERZ PRZEPIS EDYTUJ PRZEPIS USUŃ PRZEPIS

3.1.2. Strona wyszukiwania

Na stronę wyszukiwania przechodzi się naciskając przycisk „szukaj” znajdujący się w menu po lewej stronie. Na szczycie strony znajduje się wyszukiwarka z przyciskiem „Szukaj”. Naciśnięcie przycisku spowoduje filtrowanie listy wyników według naszego zapytania. Poniżej pola tekstowego znajdują się trzy możliwości wyszukiwania. Użytkownik może wyszukiwać według nazwy przepisu, autora lub składników znajdujących się w przepisie poprzez zaznaczenie jednej z trzech opcji przed naciśnięciem przycisku „Szukaj”. Po wybraniu interesującego nas przepisu poprzez naciśnięcie na jego pozycję w liście przepisów możemy skorzystać z trzech opcji (patrz punkt 3.1.1.).

KSIĄŻKA KUCHARSKA

szukaj

dodaj przepis

strona główna

wyjście

KSIĄŻKA KUCHARSKA

SZUKAJ

☒ Szukaj wg. tytułu
☐ Szukaj wg. autorów
☐ Szukaj wg. składnika

Racuchy z jabłkami / Krysia
Ciasteczka pełnoziarniste / Ania
Krupnik / Krysia
Jaglanka z bananem i sosem z masła orzechowego / Ania
Grzany cydr / Andrzej
Kokosowe musli z truskawkami / Janusz

WYBIERZ PRZEPIS **EDYTUJ PRZEPIS** **USUŃ PRZEPIS**

3.1.3. Strona dodawania/ edycji przepisu

Na tę stronę możemy zostać przekierowani poprzez naciśnięcie przycisku „dodaj” z menu głównego znajdującego się po lewej stronie. Umożliwi to dodanie nowego przepisu do bazy danych. Możemy także dostać się na tę stronę poprzez naciśnięcie przycisku „Edytuj przepis” znajdującego się na stronie głównej oraz stronie wyszukiwania. Na stronie znajdują się kolejno:

Nazwa przepisu - pole tekstowe, w które należy wpisać nazwę przepisu.

Kategoria - menu rozwijane, za pomocą którego możemy wybrać kategorię, do której chcemy przydzielić nasz przepis.

Składnik – pierwsze pole tekstowe służy do wpisania nazwy składnika. Kolejne do podania ilości w formie liczby rzeczywistej. Trzecie pole to menu rozwijane, które umożliwia wybór typu ilości, np. sztuka, kg, szklanka itd. Naciśnięcie przycisku „Dodaj” sprawi, że wpisany przez nas składnik wraz z ilością zostanie przeniesiony do listy składników, a pola służące do dodawania składnika zostaną wyczyszczone.

Dodane składniki – lista składników, które zostały dodane do przepisu. Naciśnięcie przycisku „Usuń” znajdującego się przy dodanym składniku powoduje usunięcie składnika z listy dodanych składników.

Opis przygotowania – pole tekstowe, w którym należy wpisać opis sposobu przygotowania danego dania.

Autor – pole tekstowe, w którym należy wpisać autora przepisu.

Naciśnięcie przycisku „Dodaj przepis” spowoduje wyświetlenie komunikatu „Dodano nowy przepis”, dodanie przepisu do bazy danych, oraz wyczyszczenie wszystkich pól tekstowych.

KSIAŻKA KUCHARSKA

szukaj

dodaj przepis

strona główna

wyjście

KSIAŻKA KUCHARSKA

Nazwa przepisu:

Kategoria:

Zupa

Składnik:

Szklanka

DODAJ

Dodane składniki:

Opis przygotowania:

szukaj

dodaj przepis

strona główna

wyjście

KSIĄŻKA KUCHARSKA

Dodane składniki:

Opis przygotowania:

Autor:

DODAJ PRZEPIS

szukaj

dodaj przepis

strona główna

wyjście

KSIĄŻKA KUCHARSKA

Nazwa przepisu: Ciasteczka pełnoziarniste

Kategoria: Deser

Składnik:

Szlanka

DODAJ

Dodane składniki:

Mąka pełnoziarnista, 160 g

Masło, 100 g

Miód, 2 łyżki

Cukier puder, 0,25 Szklanki

usuń

usuń

usuń

usuń

Opis przygotowania:

Masło pokroić w kostkę i utrzeć (ubić mikserem) z miodem i cukrem pudrem (oraz tahini/ zmielonym sezamem jeśli używamy). Dodać mąkę wymieszaną z proszkiem do pieczenia, dokładnie wymieszać i uformować kulę. Odrywać po kawałeczki ciasta i lepić kuleczki nieco mniejsze od orzecha włoskiego.



3.1.4. Strona wyświetlania przepisu

Użytkownik może wyświetlić tę stronę poprzez wybranie przepisu i naciśnięcie przycisku „Wybierz przepis” na stronie głównej oraz stronie wyszukiwania. Na stronie wyświetlania przepisu znajduje się nazwa przepisu, składniki potrzebne do jego zrealizowania oraz opis przygotowania. Na wysokości składników, po prawej stronie znajdują się także QR code, którego odczytanie pozwoli wyświetlić tekst przepisu na ekranie telefonu. Na dole strony znajdują się dwa przyciski: „Edytuj przepis”, który przekierowuje nas na stronę edycji przepisu, a także „Usuń przepis”, który wyświetla komunikat „Czy na pewno chcesz usunąć przepis?”. Wybranie opcji „Tak” spowoduje usunięcie wyświetlanego przepisu z bazy danych i przekierowanie użytkownika na stronę główną.

szukaj

dodaj przepis

strona główna

wyjście

KSIĄŻKA KUCHARSKA

Grzany cydr

Składniki:

- Cydr, 1 l
- Cukier trzcinowy, 5 łyżek
- Kora cynamonu, 2 Sztuki
- Gwiazdki anyżu, 2 Sztuki
- Goździki, 1 łyżka
- Zielone jabłko, 0,5 Sztuki



Opis przygotowania:

Do szerokiego większego garnka wsypać cukier trzcinowy, rozprowadzić go po całej powierzchni dna. Ustawić na ogniu i podgrzewać na średnim ogniu bez mieszania aż cukier się skarmelizuje: rozpuści, zacznie lekko się dymić i ściemnieje (będzie brązowy, ale nie...

szukaj

dodaj przepis

strona główna

wyjście

KSIĄŻKA KUCHARSKA



Opis przygotowania:

Do szerokiego większego garnka wsypać cukier trzcinowy, rozprowadzić go po całej powierzchni dna. Ustawić na ogniu i podgrzewać na średnim ogniu bez mieszania aż cukier się skarmelizuje: rozpuści, zacznie lekko się dymić i ściemnieje (będzie brązowy, ale nie czarny!).

Wówczas wlać cydr (ostrożnie, będzie pryskać) i wymieszać. Dodać cynamon, goździki i anyż i podgrzewać na średnim ogniu aż napój będzie gorący (ale nie zagotowany), przez około 5 - 6 minut. W międzyczasie dodać cząstki jabłka.

Przełać do szklanek i podawać na gorąco.

EDYTUJ PRZEPIS

USUŃ PRZEPIS

3.2. Komunikaty

„Nie udało się połączyć z bazą” – komunikat występuje, gdy plik CookingBook.sqlite nie znajduje się w odpowiednim miejscu. Aby temu zaradzić, należy umieścić go w tym samym miejscu, co plik App.config. Może on także występować, jeżeli aplikacja nie została zrebuildowana po pobraniu projektu.

„Nie wybrano żadnego przepisu” – komunikat ten wyświetla się podczas prób wyświetlenia, edycji lub usunięcia przepisu, podczas gdy żaden przepis z listy nie został wybrany. Aby komunikat się nie wyświetlał, należy wybrać jeden z przepisów z listy poprzez kliknięcie na jego nazwę.

„Czy na pewno chcesz usunąć przepis?” – zapytanie pojawi się, gdy użytkownik naciśnie przycisk „Usuń przepis” przy wybranym przepisie z listy. Jeśli użytkownik wybierze opcję „Tak”, przepis zostanie usunięty z bazy danych.

„Uzupełnij nazwę składnika” – komunikat jest wyświetlany, gdy użytkownik w trakcie tworzenia nowego przepisu podejmuje próbę dodania do listy składników składnika, którego nazwa jest pusta. Należy uzupełnić pole tekstowe nazwą składnika, który chcemy dodać.

„Uzupełnij puste pola” – komunikat jest wyświetlany, gdy użytkownik próbuje dodać do bazy nowy przepis, nie uzupełniając pola opisu przygotowania lub nazwy przepisu. Aby komunikat nie był wyświetlany, należy uzupełnić puste pola.

„Czy chcesz dodać przepis bez autora?” – zapytanie pojawia się, gdy pozostawimy puste pole tekstowe, w którym powinien być wpisany autor. Jeżeli użytkownik wybierze opcję „Tak”, to w miejscu autora przepisu będzie wyświetlał się napis „Anonimowy”.

„Czy na pewno chcesz dodać przepis nie zawierający składników?” – zapytanie pojawia się, gdy użytkownik dodaje do bazy przepis z pustą listą składników. Jeżeli użytkownik wybierze opcję „Tak”, przepis zostanie dodany do bazy.

„Dodano nowy przepis” – komunikat wyświetlany jest po poprawnym dodaniu nowego przepisu do bazy danych.

„Przepis został zedytowany” – komunikat wyświetlany jest po poprawnym zmodyfikowaniu przepisu znajdującego się w bazie danych.

„Przepis został usunięty” – komunikat wyświetli się, gdy usuniemy przepis z bazy danych za pomocą przycisku „Usuń przepis” znajdującego się na stronie wyświetlającej przepis.

4. Specyfikacja wewnętrzna

4.1. Klasy

4.1.1. Author

Pola klasy:

<code>public long ID</code>	ID autora
<code>public string Name</code>	Nazwa autora
<code>public virtual ICollection<Recipe> Recipes</code>	Kolekcja przepisów danego autora

4.1.2. Ingredient

Pola klasy:

<code>public long ID</code>	ID składnika
<code>public string Name</code>	nazwa składnika
<code>public Nullable<double> Quantity</code>	ilość składnika
<code>public Nullable<long> RecipeID</code>	Klucz obcy, ID przepisu
<code>public Nullable<long>TypeID</code>	Klucz obcy, ID typu ilości
<code>public virtual QuantityType QuantityType</code>	Leniwe powiązanie z klasą QuantityType
<code>public virtual Recipe Recipe</code>	Leniwe powiązanie z klasą Recipe

Metody:

<code>public override string ToString()</code>	Metoda, która ma za zadanie odmienić poprawnie typ ilości w zależności od tego, jaką ilość składnika podał użytkownik. Metoda zwraca łańcuch znaków w postaci: [Nazwa składnika], [Ilość] [Typ ilości]
--	---

4.1.3. QuantityType

Pola klasy:

<code>public long ID</code>	ID typu ilości składnika
<code>public string Name</code>	Nazwa typu ilości składnika
<code>public virtual ICollection<Ingredient> Ingredients</code>	kolekcja składników danego typu ilości

4.1.4. Recipe

Pola klasy:

<code>public long ID</code>	ID przepisu
<code>public string Title</code>	Nazwa przepisu
<code>public string Description</code>	opis przygotowania dla przepisu
<code>public Nullable<long> AuthorID</code>	Klucz obcy, ID autora przepisu
<code>public Nullable<long> RecipeCategoryID</code>	Klucz obcy, ID kategorii przepisu
<code>public virtual Author Author</code>	Leniwe powiązanie z klasą Author
<code>public virtual ICollection<Ingredient> Ingredients</code>	Kolekcja składników danego przepisu
<code>public virtual RecipeCategory RecipeCategory</code>	Leniwe powiązanie z klasą RecipeCategory

Metody:

<code>public override string ToString()</code>	Metoda zwraca łańcuch znaków o postaci: [Nazwa przepisu] / [Autor]
--	---

4.1.5. RecipeCategory

Pola klasy:

<code>public long ID</code>	ID kategorii przepisu
<code>public string Name</code>	nazwa kategorii przepisu
<code>public virtual ICollection<Recipe> Recipes</code>	Kolekcja przepisów danej kategorii przepisu

Metody:

<code>public override string ToString()</code>	Metoda zwraca nazwę kategorii przepisu
--	--

4.1.6. AddRecipePageViewModel

View Model dla modelu AddRecipePage.xaml

Pola klasy:

<code>public ICommand AddIngredientCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „dodaj”
<code>public ICommand AddRecipeCmd</code>	Interfejs do obsługi zdarzenia

	kliknięcia na przycisk „dodaj przepis”
<code>public ICommand DeleteIngredientCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „usuń”
<code>public ICommand EditRecipeCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „edytuj przepis”
<code>public ObservableCollection<RecipeCategory> CategoriesCB</code>	Kolekcja kategorii przepisów
<code>public ObservableCollection<Ingredient> IngredientList</code>	Kolekcja dodanych składników
<code>public ObservableCollection<string> QuantityCB</code>	Kolekcja typów ilości składnika
<code>public string AuthorTxt</code>	Właściwość odpowiadająca nazwie autora na widoku
<code>public string DescriptionTxt</code>	Właściwość odpowiadająca opisowi przygotowania przepisu na widoku
<code>public string IngredientNameTxt</code>	Właściwość odpowiadająca nazwie składnika na widoku
<code>public string IngredientQuantityTxt</code>	Właściwość odpowiadająca ilości składnika na widoku
<code>public string TitleTxt</code>	Właściwość odpowiadająca nazwie przepisu na widoku
<code>public string SelectedQuantityCB</code>	Właściwość odpowiadająca nazwie wybranego typu ilości na widoku
<code>public bool IsAdd</code>	Właściwość określająca czy dany widok jest w trybie dodawania
<code>public bool IsEdit</code>	Właściwość określająca czy dany widok jest w trybie edycji
<code>public RecipeCategory SelectedCategoryCB</code>	Właściwość przechowująca obecnie zaznaczoną kategorię przepisu
<code>public int SelectedCategoryIndex</code>	Właściwość przechowująca ID kategorii przepisu
<code>private long RecipeID</code>	Właściwość przechowująca ID przepisu

Metody:

<code>public AddRecipePageViewModel()</code>	Konstruktor klasy
<code>public void Fill(long id)</code>	Metoda mająca za zadanie wypełnienie widoku AddRecipePage.xaml oraz przejście do trybu edycji. Metoda przyjmuje ID przepisu przeznaczonego do edycji
<code>private void AddIngredient()</code>	Metoda mająca za zadanie dodać składnik do listy składników po naciśnięciu przycisku „Dodaj”

<code>private void AddRecipe()</code>	Metoda ma za zadanie dodać nowy przepis do bazy danych po naciśnięciu przycisku „Dodaj przepis”.
<code>public void ClearPage()</code>	Metoda czyści wszystkie pola tekstowe oraz ustawia pola wyboru na pozycje domyślne.
<code>private void DeleteIngredient(object ingredient)</code>	Metoda ma za zadanie usunąć składnik z listy składników po naciśnięciu przycisku „Usuń”. Metoda przyjmuje obiekt składnika do usunięcia
<code>private void EditRecipe()</code>	Metoda ma za zadanie zapisać do bazy danych zedytowany przepis po naciśnięciu przycisku „Edytuj przepis”
<code>private long GetAuthorID()</code>	Metoda ma za zadanie znaleźć w bazie danych ID autora, którego nazwa jest wpisana w polu tekstowym, a następnie je zwrócić. Jeżeli nazwa autora nie zostanie odnaleziona, metoda zwraca -1.
<code>virtual protected void OnPropertyChanged(string propName)</code>	Metoda ta ma za zadanie wyzwolenie zdarzenia PropertyChanged i przekazanie nazwy właściwości. Prowadzi to do odświeżenia widoku.

4.1.7. HomePageViewModel

View Model dla modelu HomePage.xaml

Pola klasy:

<code>public ICommand AllRecipesCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „wszystkie”
<code>public ICommand OtherCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „inne”
<code>public ICommand BreakfastCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „śniadania”
<code>public ICommand DeleteCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „usuń przepis”
<code>public ICommand DessertCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „desery”
<code>public ICommand EditCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „edytuj przepis”
<code>public ICommand MainDishCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „dania główne”
<code>public ICommand SearchCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „szukaj”
<code>public ICommand ShowCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „wybierz

	przepis”
<code>public ICommand SoupCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „zupy”
<code>public string Page</code>	Właściwość typu string zawierająca ścieżkę do odpowiednich widoków
<code>public string SearchTxt</code>	Właściwość odpowiadająca wyszukiwanej frazie na widoku
<code>public ObservableCollection<Recipe> RecipeList</code>	Kolekcja przepisów
<code>public Recipe SelectedRecipe</code>	Właściwość przechowująca obecnie zaznaczony przepis

Metody:

<code>public HomePageViewModel()</code>	Konstruktor klasy
<code>public void Refresh()</code>	Metoda mająca za zadanie odświeżenie listy przepisów.
<code>private void ShowBreakfast()</code>	Metoda ma za zadanie wyświetlać na liście przepisów tylko przepisy należące do kategorii „Śniadanie”. Jest ona wywoływana po naciśnięciu przycisku „Śniadania”.
<code>private void Delete()</code>	Metoda usuwająca wybrany przez użytkownika przepis z bazy danych. Jest wywoływana po naciśnięciu przycisku „Usuń przepis”.
<code>private void Edit()</code>	Metoda przechodząca do strony edycji dla wybranego przepisu. Jest wywoływana po naciśnięciu przycisku „Edytuj przepis”.
<code>private void Search()</code>	Metoda mająca za zadanie wyświetlić na liście przepisów tylko przepisy spełniające kryterium wyszukiwania zawarte w polu tekstowym SearchTxt. Metoda jest wywoływana po naciśnięciu przycisku „Szukaj”.
<code>private void Show()</code>	Metoda przechodząca do strony wyświetlania wybranego przepisu. Jest wywoływana po naciśnięciu przycisku „Wybierz przepis”.
<code>private void ShowAll()</code>	Metoda ma za zadanie wyświetlać na liście przepisów wszystkie przepisy. Jest ona wywoływana po naciśnięciu przycisku „Wszystkie”.
<code>private void ShowOther()</code>	Metoda ma za zadanie wyświetlać na liście przepisów tylko przepisy należące do kategorii „Inne”. Jest ona wywoływana po naciśnięciu przycisku „Inne”.
<code>private void ShowDessert()</code>	Metoda ma za zadanie wyświetlać na

	liście przepisów tylko przepisy należące do kategorii „Deser”. Jest ona wywoływana po naciśnięciu przycisku „Desery”.
<code>private void ShowMainDish()</code>	Metoda ma za zadanie wyświetlać na liście przepisów tylko przepisy należące do kategorii „Danie główne”. Jest ona wywoływana po naciśnięciu przycisku „Dania główne”.
<code>private void ShowSoup()</code>	Metoda ma za zadanie wyświetlać na liście przepisów tylko przepisy należące do kategorii „Zupa”. Jest ona wywoływana po naciśnięciu przycisku „Zupy”.
<code>virtual protected void OnPropertyChanged(string propName)</code>	Metoda ta ma za zadanie wyzwolenie zdarzenia PropertyChanged i przekazanie nazwy właściwości. Prowadzi to do odświeżenia widoku.

4.1.8. MainViewModel

View Model dla modelu MainWindow.xaml

Pola klasy:

<code>public ICommand AddCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „dodaj”
<code>public ICommand CloseCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „wyjście”
<code>public ICommand HomeCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „strona główna”
<code>public ICommand SearchCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „szukaj”
<code>public string Page</code>	Właściwość typu string zawierająca ścieżkę do odpowiednich widoków

Metody:

<code>public MainViewModel()</code>	Konstruktor klasy.
<code>public void Add()</code>	Metoda przechodząca do widoku dodawania przepisu. Jest wywoływana po naciśnięciu przycisku „dodaj przepis”.
<code>public void Home()</code>	Metoda przechodząca do widoku strony głównej. Jest wywoływana po naciśnięciu przycisku „strona główna”.
<code>private void Close()</code>	Metoda mająca za zadanie zamknąć

	aplikację. Jest wywoływana po naciśnięciu przycisku „wyjście”.
<code>private void Search()</code>	Metoda przechodząca do widoku strony głównej. Jest wywoływana po naciśnięciu przycisku „szukaj”.
<code>virtual protected void OnPropertyChanged(string propName)</code>	Metoda ta ma za zadanie wyzwolenie zdarzenia PropertyChanged i przekazanie nazwy właściwości. Prowadzi to do odświeżenia widoku.

4.1.9. SearchPageViewModel

View Model dla modelu SearchPage.xaml

Pola klasy:

<code>public ICommand DeleteCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „usuń przepis”
<code>public ICommand EditCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „edytuj przepis”
<code>public ICommand SearchCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „szukaj”
<code>public ICommand ShowCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „wybierz przepis”
<code>public string SearchTxt</code>	Właściwość odpowiadająca wyszukiwanej frazie na widoku
<code>public bool SelectedAuthor</code>	Właściwość określająca kryterium wyszukiwania według autora
<code>public bool SelectedIngredient</code>	Właściwość określająca kryterium wyszukiwania według składników przepisu
<code>public bool SelectedTitle</code>	Właściwość określająca kryterium wyszukiwania według nazwy przepisu
<code>public Recipe SelectedRecipe</code>	Właściwość przechowująca obecnie zaznaczony przepis
<code>public ObservableCollection<Recipe> RecipeList</code>	Kolekcja przepisów

Metody:

<code>public SearchPageViewModel()</code>	Konstruktor klasy
<code>public void Refresh()</code>	Metoda mająca za zadanie odświeżenie listy przepisów.
<code>private void Delete()</code>	Metoda usuwająca wybrany przez użytkownika przepis z bazy danych.

	Jest wywoływana po naciśnięciu przycisku „Usuń przepis”.
<code>private void Edit()</code>	Metoda przechodząca do strony edycji dla wybranego przepisu. Jest wywoływana po naciśnięciu przycisku „Edytuj przepis”.
<code>private void Search()</code>	Metoda mająca za zadanie wyświetlić na liście przepisów tylko przepisy spełniające kryterium wyszukiwania zawarte w polu tekstowym SearchTxt. Metoda jest wywoływana po naciśnięciu przycisku „Szukaj”.
<code>private void Show()</code>	Metoda przechodząca do strony wyświetlania wybranego przepisu. Jest wywoływana po naciśnięciu przycisku „Wybierz przepis”.
<code>virtual protected void OnPropertyChanged(string propName)</code>	Metoda ta ma za zadanie wyzwolenie zdarzenia PropertyChanged i przekazanie nazwy właściwości. Prowadzi to do odświeżenia widoku.

4.1.10. SharedVM

Klasa SharedVM ma za zadanie przechowywać obiekty View Modeli.

Pola klasy:

<code>public static HomePageViewModel HomeVM</code>	Statyczna klasa HomePageViewModel
<code>public static MainViewModel MainVM</code>	Statyczna klasa MainViewModel
<code>public static AddRecipePageViewModel RecipeVM</code>	Statyczna klasa AddRecipePageViewModel
<code>public static SearchPageViewModel SearchVM</code>	Statyczna klasa SearchPageViewModel

Metody:

<code>public SharedVM()</code>	Konstruktor klasy.
--------------------------------	--------------------

4.1.11. ShowRecipePageViewModel

View Model dla modelu ShowRecipePage.xaml

Pola klasy:

<code>public ICommand DeleteRecipeCmd</code>	Interfejs do obsługi zdarzenia kliknięcia na przycisk „usuń przepis”
<code>public ICommand EditRecipeCmd</code>	Interfejs do obsługi zdarzenia kliknięcia

	na przycisk „edytuj przepis”
<code>public BitmapImage QrCode</code>	Bitmapa zawierająca QR code przepisu
<code>public string RecipeDescription</code>	Właściwość prezentująca opis przepisu na widoku
<code>public string RecipeIngredients</code>	Właściwość prezentująca składniki przepisu na widoku
<code>public string RecipeTitle</code>	Właściwość prezentująca nazwę przepisu na widoku
<code>public Recipe SelectedRecipe</code>	Właściwość przechowująca wybrany przez użytkownika przepis z listy

Metody:

<code>public ShowRecipePageViewModel(long recipeID)</code>	Konstruktor klasy przyjmujący ID przepisu wybranego przez użytkownika.
<code>private void Delete()</code>	Metoda usuwająca wyświetlany przepis z bazy danych. Jest wywoływana po naciśnięciu przycisku „Usuń przepis”.
<code>private void Edit()</code>	Metoda przechodząca do strony edycji dla wyświetlanego przepisu. Jest wywoływana po naciśnięciu przycisku „Edytuj przepis”.
<code>private void setQRData()</code>	Metoda mająca za zadanie utworzyć QR code dla wyświetlanego przepisu.
<code>virtual protected void OnPropertyChanged(string propName)</code>	Metoda ta ma za zadanie wyzwolenie zdarzenia PropertyChanged i przekazanie nazwy właściwości. Prowadzi to do odświeżenia widoku.

4.1.12. NumberValidationRule

Klasa służąca do walidacji pola tekstowego na stronie dodawania przepisu, w które wpisuje się ilość składnika.

Metody:

<code>public override ValidationResult Validate(object value, System.Globalization.CultureInfo cultureInfo)</code>	Metoda ma za zadanie sprawdzać, czy w polu ilości składnika znajduje się liczba rzeczywista. Jeżeli nie, wyświetlany jest komunikat: „Zły format liczby”.
--	---

4.1.13. RelayCommand

Klasa skopiowana ze strony:

<https://msdn.microsoft.com/pl-pl/library/wprowadzenie-do-wzorca-projektowego-model-view-viewmodel-na-przykladzie-aplikacji-wpf.aspx>

Jest to klasa implementująca komendę (ICommand).