# Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej



FoodNET

Aplikacja sugerująca przepisy kulinarne na podstawie stanu lodówki

Dokumentacja

Adam Gryz

Damian Baciur

# 1. Specyfikacja

# 1.1 Opis biznesowy

Niniejsza aplikacja może posłużyć jako codzienny kuchenny pomocnik . Głównym zadaniem aplikacji jest pomoc użytkownikowi w znalezieniu przepisów kulinarnych, które można przyrządzić przy aktualnym stanie jego lodówki. System umożliwia zarządzanie wirtualną lodówką, poprzez edytowalną listę produktów. Ponadto użytkownik ma możliwość przeglądania, dodawania, edytowania i usuwania swoich przepisów, które służą jako baza do przeszukiwań.

# 1.2 Wymagania funkcjonalne (User stories)

**2.1. Jako użytkownik mam możliwość zarządzania wirtualną lodówką**

Użytkownik aplikacji ma możliwość zarządzania wirtualną lodówką. Ma możliwość dodawania i usuwania produktów (nazwa produktu, kategoria). Od początku aplikacja będzie posiadać dane o podstawowych produktach i odpowiadające im kategorie.

**2.2. Jako użytkownik mam możliwość zarządzania** **przepisami.**

Użytkownik może dodawać, edytować i usuwać przepisy kulinarne ( nazwa, zdjęcie, lista produktów, treść przepisu). Przy dodawaniu przepisów może korzystać z produktów już znadujących się w bazie danych lub dodać nowe produkty.

**2.3.** **Jako użytkownik mam możliwość znalezienia odpowiednich przepisów**

Po wybraniu ewentualnych dodatkowych kryteriów związanych z przepisem, użytkownik może przeglądać listę przepisów, które można przygotować z produktów aktualnie znajdujących się w lodówce.

# 1.3 Wybrane technologie

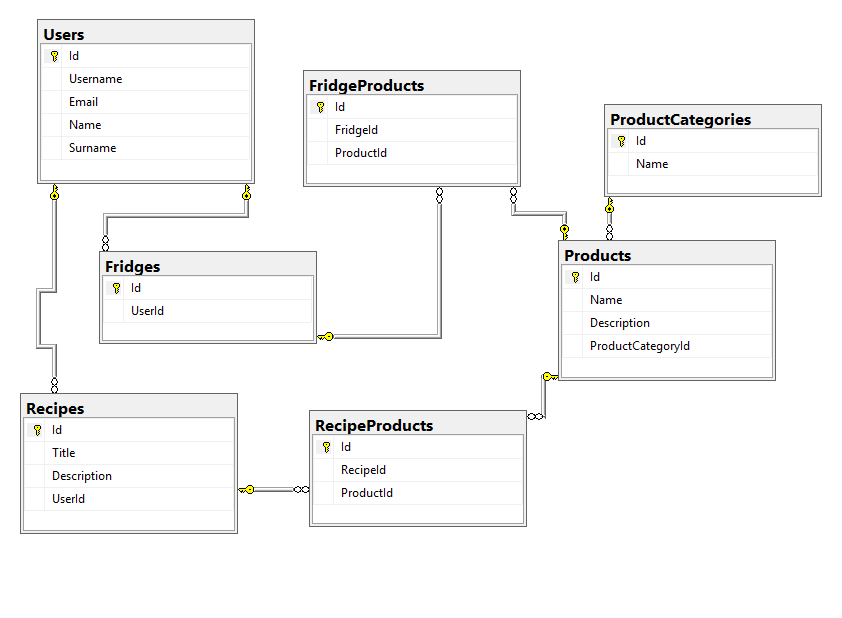
**3.1.System zarządzania bazą danych:** Microsoft SQL Server

**3.2.Narzędzie ORM:** Entity Framework

**3.3.Narzędzie do tworzenia GUI:** WPF + Prism

**3.4.Pozostałe :** Autofac, Git, SourceTree, BitBucket

# 1.4 Schemat bazy danych



**Pola wymagane w poszczególnych tabelach:**

|  |  |
| --- | --- |
| Nazwa tabeli | Wymagane pola |
| Users | UserID, Email, Password, FridgeID |
| Fridges | UserID, FridgeID |
| Recipes | RecipeID, Name, Content |
| Products | ProductID, Name, CategoryID |
| FridgesProducts | FridgeID, ProductID |
| RecipesProducts | RecipeID, ProductID, Essential |
| ProductCategories | CategoryID, Name |

**Walidacja wartości w bazie:**

|  |  |
| --- | --- |
| Nazwa pola | Walidacja |
| Users.Email | unikatowy w obrębie całej tabeli, zgodny z ogólnie przyjętym formatem |
| Users.Name | składa się wyłącznie z liter |
| Users.Surname | składa się wyłącznie z liter |
| Product.CategoryID | wartość równa id istniejącego rekordu w ProductCategories |

# 1.5 Harmonogram

**16.10.2017** - wybór używanych technologii

**23.10.2017 -** zakończenie i przedstawienie prowadzącemu specyfikacji

**25.10.2017 –** stworzenie modelu i bazy danych

**27.10.2017 –** ukończenie podstawowej logiki interfejsu

**01.11.2017 –**  ukończenie prac związanych z wyglądem interfejsu

**06.11.2017** – widok i logika rezultatu aplikacji

**09.11.2017 –** testy, poprawki wizualne interfejsu, refaktoryzacja kodu

**11.11.2017** – zakończenie fazy projektu

# 2. Dokumentacja powykonawcza

# Wymagania systemowe

1. System operacyjny:

a) Windows 7 SP1 (x86 i x64)  
b) Windows 8 (x86 i x64)  
c) Windows 8.1 (x86 i x64)  
d) Windows 10

e) Windows Server 2008 R2 SP1 (x64)

f) Windows Server 2012 (x64)

g) Windows Server 2012 R2 (x64)

2. Platforma Microsoft .NET Framework w wersji 4.6.1 lub wyższej

# 3. Baza danych SQL Server Express wersja 2008 R2 lub wyższa

# 2.2 Wykorzystane biblioteki zewnętrzne

|  |  |  |  |
| --- | --- | --- | --- |
| **nr.** | **Komponent i wersja** | **Opis** | **Licencja** |
| **1.** | Autofac.dll v.4.6.2.0 | Kontener Inversion of Control dla aplikacji w środowisku .NET | MIT |
| **2.** | MaterialDesignColors.dll  v.1.1.3  MaterialDesignThemes.dll  v.2.3.1.953  MaterialDesignThemes.MahApps.dll v.0.0.11 | Biblioteki zawierające templatki i style Material Design dla kontrolek WPF | MS-PL |
| **3.** | xunit.core.dll  xunit.assert.dll  xunit.abstractions.dll  xunit.execution.desktop.dll | Narzędzie do tworzenia testów jednostkowych na platformy .NET | Apache License 2.0 |
| **4.** | Moq.dll | Framework do mockowania na platformy .NET | MIT |

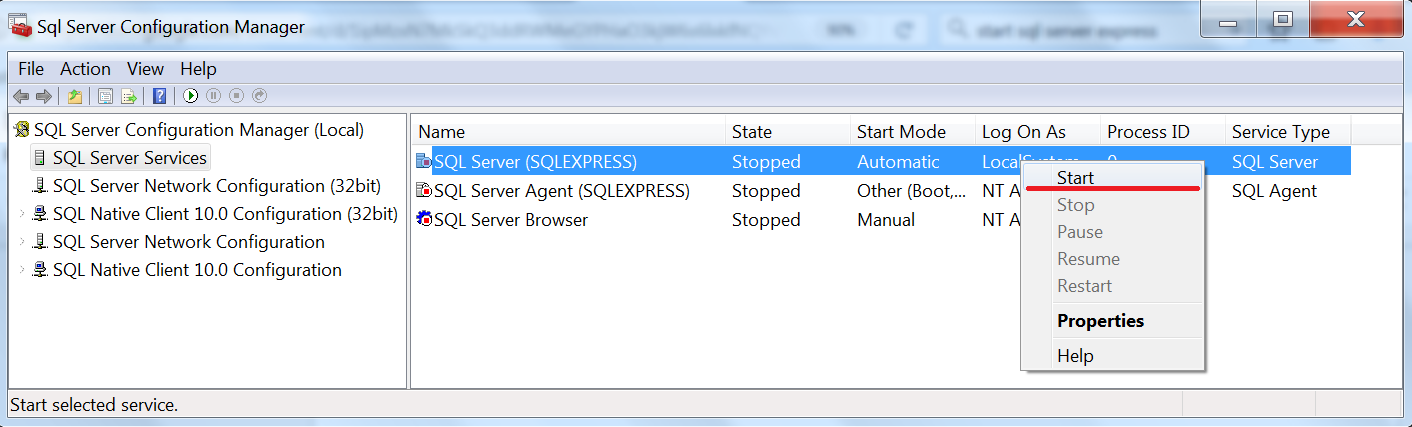
2.3 Instrukcja instalacji

**1. Instalacja .NET Framework 4.6.1**  
Platformę .NET instalujemy z oficjalnej strony z linkami do pobierania Microsoftu. Obecnie podana wersja dostępna jest pod [link](https://www.microsoft.com/pl-pl/download/details.aspx?id=49981) (https://www.microsoft.com/pl-pl/download/details.aspx?id=49981). W przypadku wygaśnięcia linku wyszukujemy “.NET Framework 4.6.1” i korzystamy z nawigacji na stronie Microsoftu. Dalej postępujemy z instrukcją instalacji. W przypadku nieznalezienia danej wersji, wybieramy dowolną o wyższym numerze.

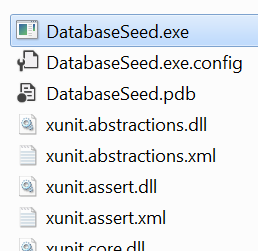
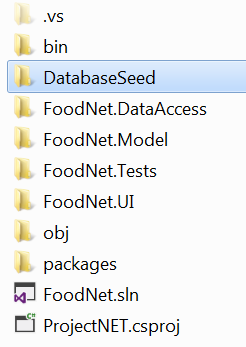
**2.Instalacja SQL Server Express**  
Podobnie jak w punkcie 1, korzystamy z oficjalnych stron do pobierania Microsoftu. Obecnie sugerowaną wersją jest SQL Server 2017 Express dostępny pod[link](https://www.microsoft.com/pl-pl/sql-server/sql-server-editions-express) (https://www.microsoft.com/pl-pl/sql-server/sql-server-editions-express). Dalej postępujemy zgodnie z instrukcją instalacji.

# 2.4 Instrukcja uruchomienia/utrzymania

**1.Konfiugracja SQL Servera**Przy pierwszym uruchomieniu aplikacji musimy wystartować pobrany wcześniej serwer SQL. Startujemy go jako Service, dzięki czemu działa w tle i nie musimy więcej zajmować się jego uruchamianiem. W tym celu otwieramy menu **Start**, dalej **All Programs** i wybieramy **Microsoft SQL Server [wybrana wersja].** Rozwijamy folder, przechodzimy do **Configuration Tools** i startujemy **SQL Server Configuration Manager** zgodnie z przykładem na rysunku. W razie problemów korzystamy ze wsparcia ze stron Microsoftu, obecnie pod [link](https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/start-stop-pause-resume-restart-sql-server-services) (<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/start-stop-pause-resume-restart-sql-server-services>).

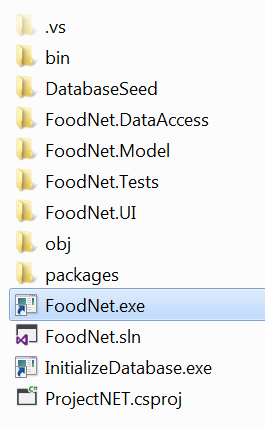
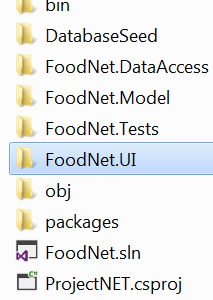


**2.Inicjalizacja bazy danych**Aby w pełni skorzystać z funkcjonalności aplikacji *FoodNet* powinniśmy zainicjalizować bazę danych danymi zebranymi przy projektowaniu aplikacji. Dzięki temu krokowi będziemy mieli dostęp do wielu predefiniowanych produktów i przepisów. W tym celu wystarczy uruchomić plik DatabaseSeed.exe w podfolderze **DatabaseSeed/bin/Debug/** folderu głównego pobranej aplikacji.



**UWAGA:** Inicjalizacja bazy danych powinno być wykonania **tylko raz** przed pierwszym użyciem aplikacji. Za każdym razem **wszystkie** wprowadzone przez użytkownika zmiany zostaną stracone.

**3.Uruchomienie aplikacji**Aplikację uruchamiamy z podfolderu **FoodNet.UI/bin/Debug/** folderu głównego uruchamiając plik FoodNet.exe.



# 2.5 Instrukcja użytkownika

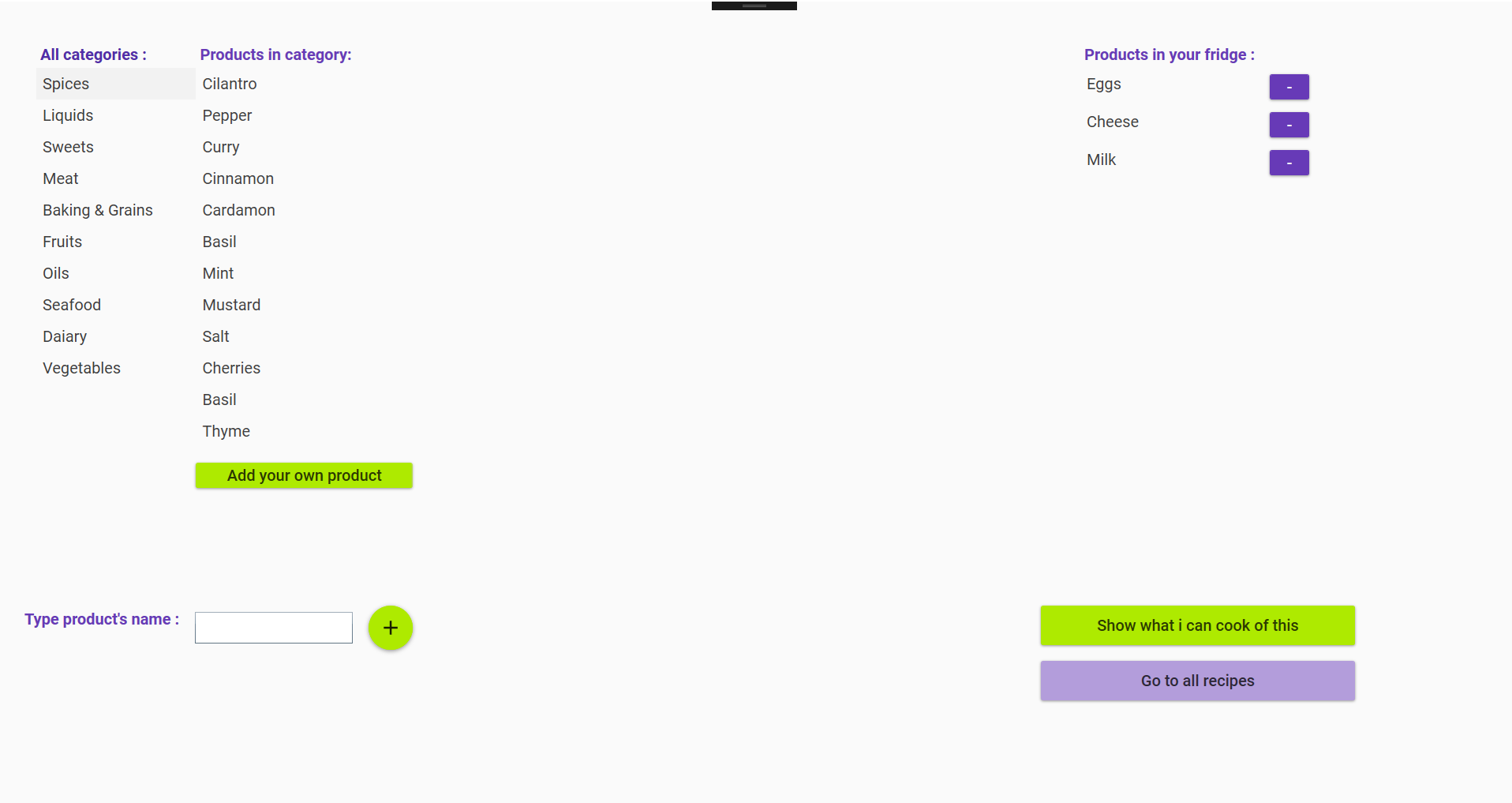
**Ekran startowy**

Umożliwia nawigację do lodówki i księgi przepisów.



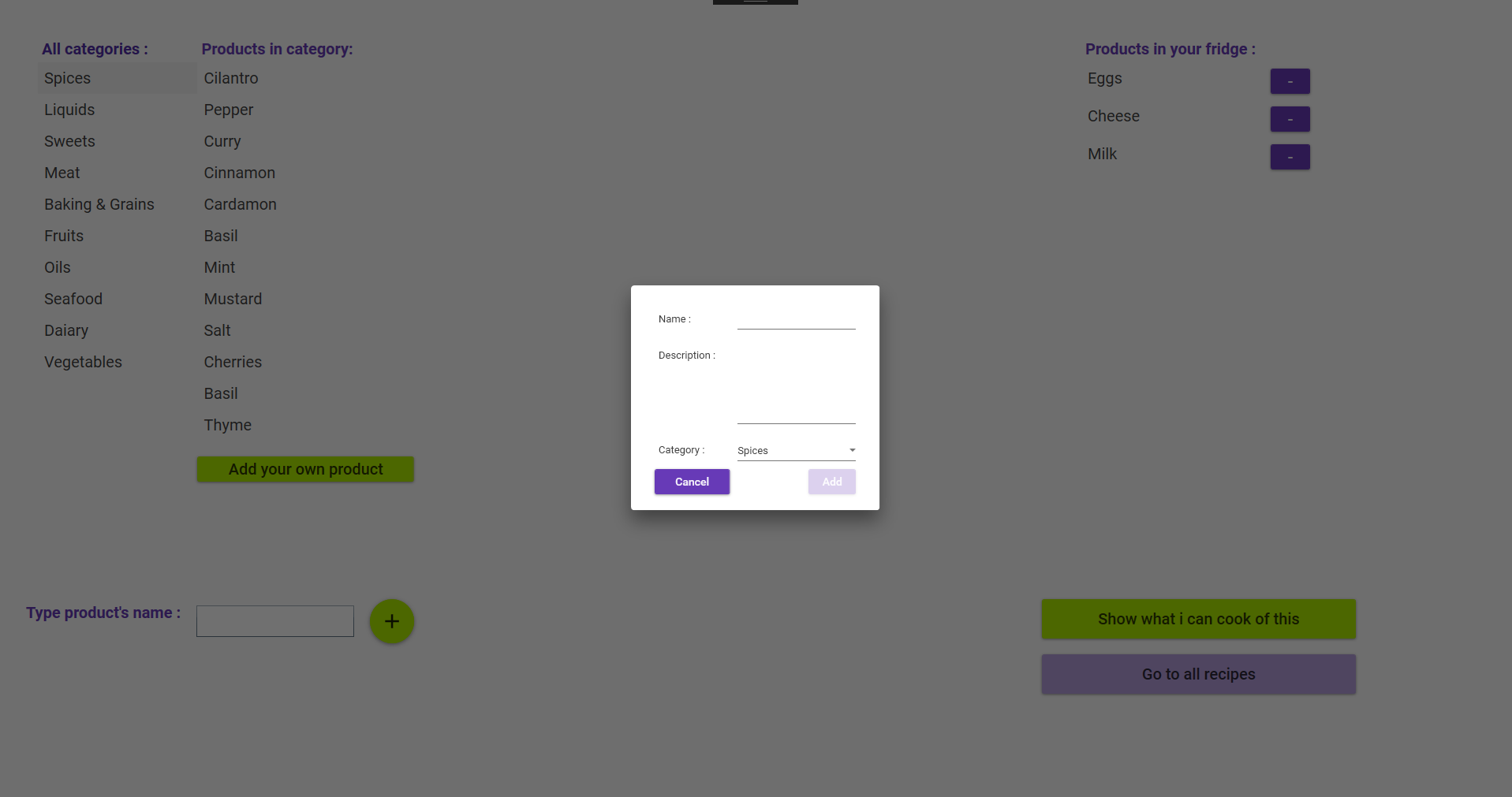
**Ekran lodówki**

* Klikając na określoną kategorię użytkownik może wyświetlić produkty do niej należące.
* Klikając na nazwę produktu, użytkownik może dodać produkt do lodówki
* Używając przycisku „Add your own product” użytkownik może dodać nowy produkt do dowolnej kategorii (domyślnie do aktualnie oglądanej kategorii)
* Użytkownik może także skorzystać z pola tekstowego, które podpowiada nazwy produktów, po kliknięciu na przycisk z plusem, może dodać produkt do lodówki
* Przy pomocy przycisków z minusami użytkownik może usunąć produkt, któremu odpowiada przycisk
* Użytkownik ma także do dyspozycji dwa przyciski nawigacyjne : jeden przeniesie go do listy przepisów, które może wykonać na podstawie stanu lodówki, drugi przeniesie go do wszystkich przepisów



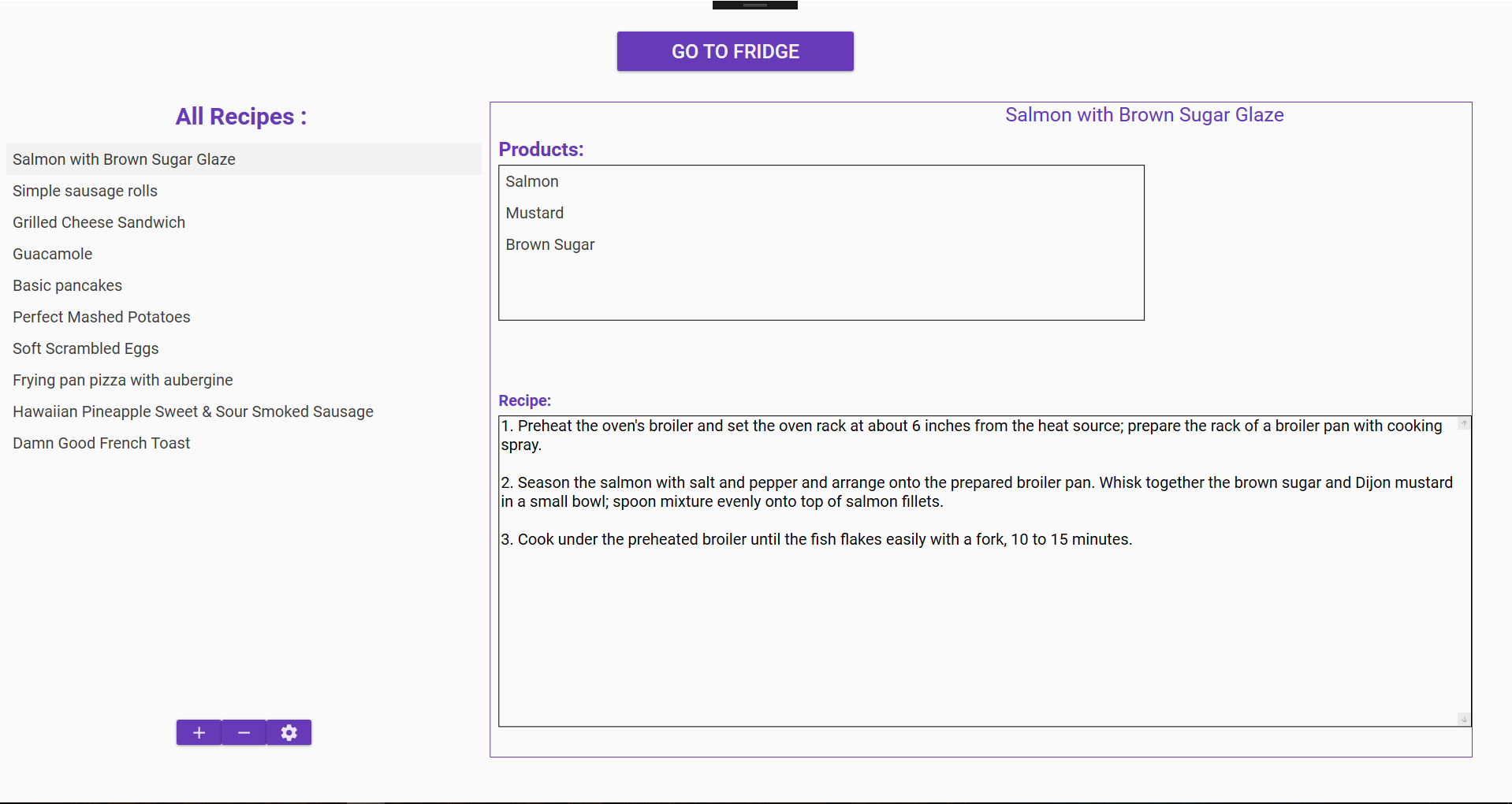
**Okno dialogowe nowego produktu**

* Użytkownik może wprowadzić nazwę produktu, opis i kategorię
* Dopóki nie będzie wybrana kategoria i nie będzie podana nazwa produktu, nie jest możliwe dodanie produktu
* Takie samo okno dialogowe znajduje się w widokach przepisów



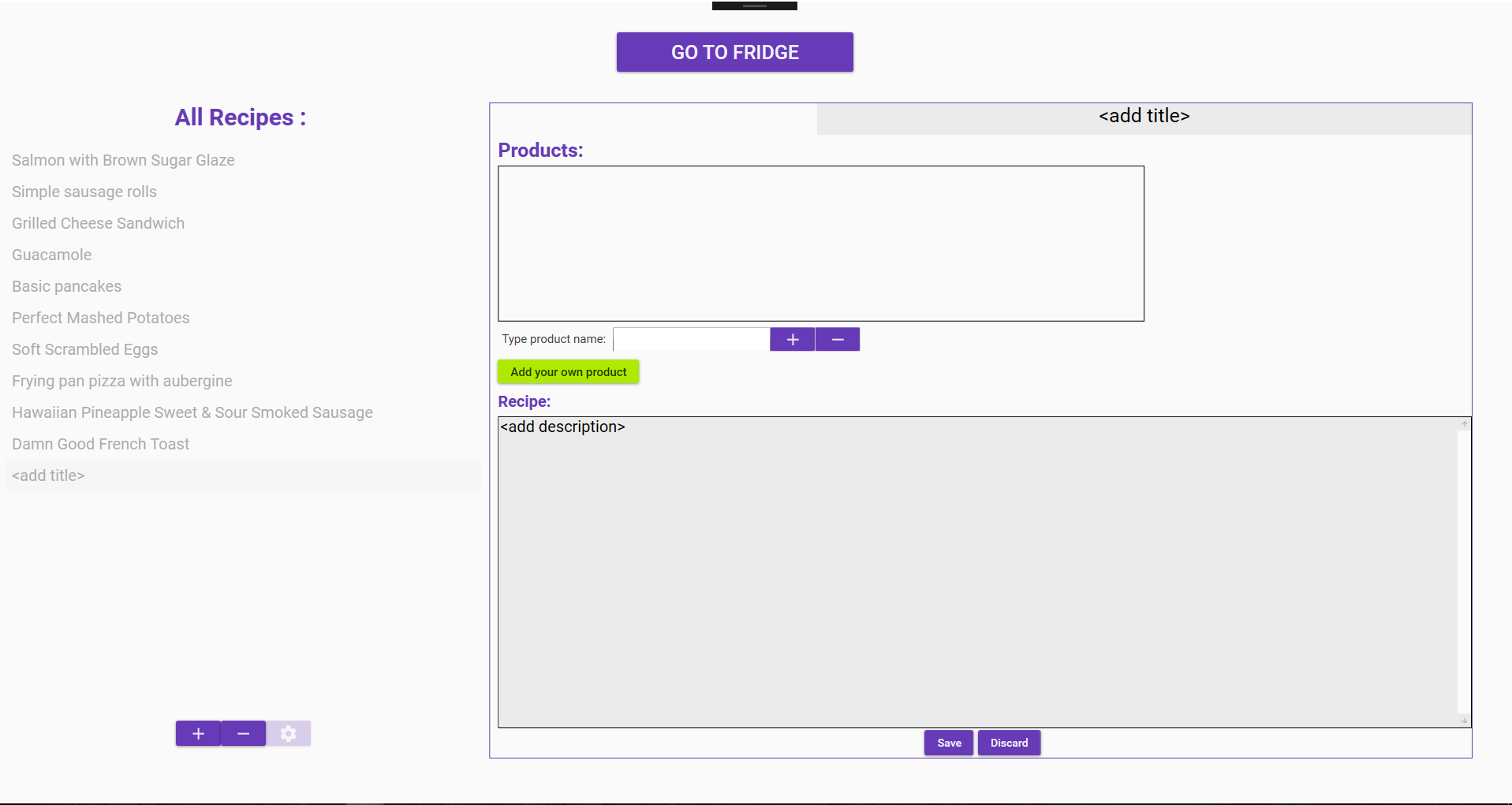
**Ekran wszystkich przepisów**

* Użytkownik może wybrać interesujący go przepis, którego detale zobaczy po prawej stronie ekranu
* Edycja przepisów jest zablokowana do momentu wybrania któregoś z przepisów i naciśnięcia przycisku 
* Użytkownik może dodać nowy przepis naciskając , treść nowego przepisu będzie edytowalna w panelu po prawej stronie
* Użytkownik może usunąć przepis , ale tylko wtedy gdy jakiś jest zaznaczony



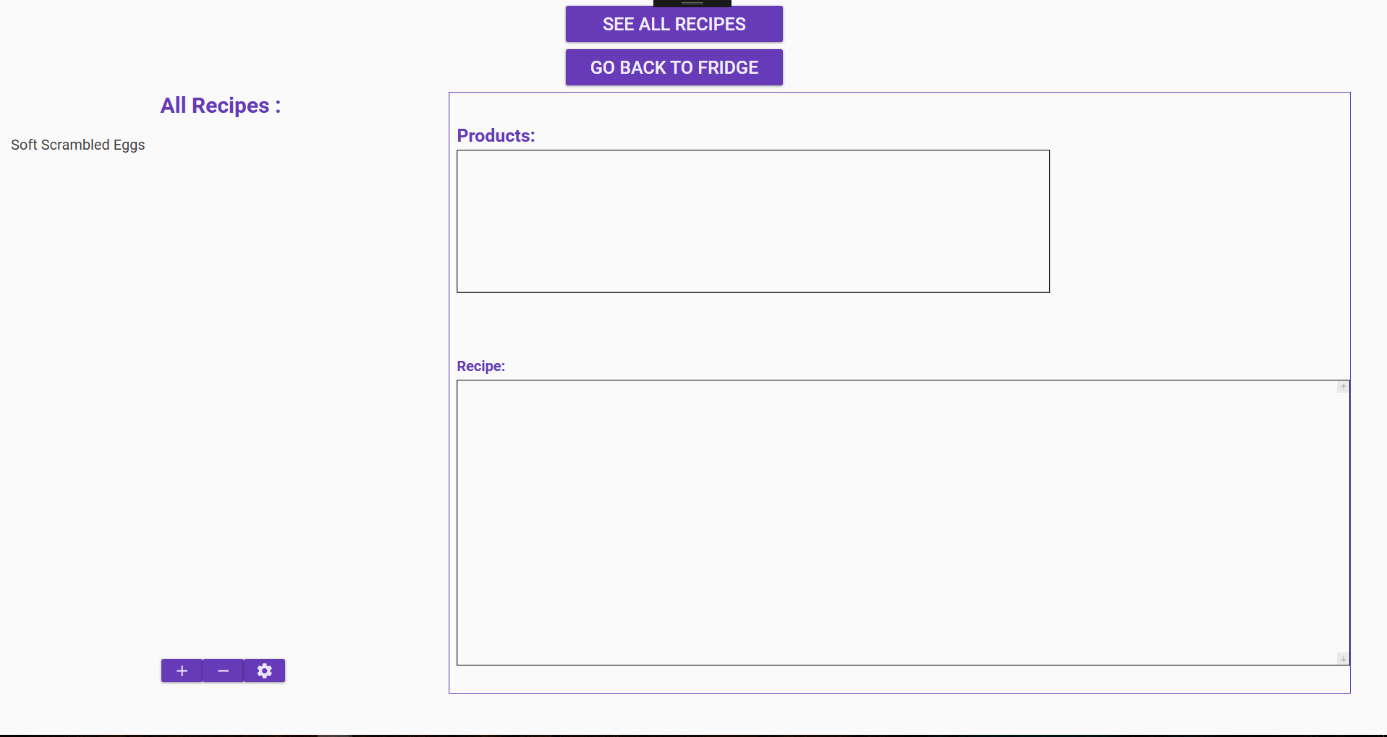
**Tryb dodawania/edycji przepisów**

* Po dodaniu nowego przepisu edytowalne pola tekstowe oznaczone są słownie i szarym tłem
* Wybierając produkt z listy możemy go usunąć przyciskiem 
* Możemy również dodawać produkty z listy lub tworzyć nowe tak samo jak na ekranie lodówki i produktów
* Wprowadzone zmiany możemy zapisać lub odrzucić
* Z ekranu przepisów możemy przyciskiem u góry udać się do ekranu lodówki



**Ekran dostępnych przepisów**

* Ekran dostępnych przepisów działa identycznie do ekranu wszystkich przepisów, inny jest jedynie jego zestaw przepisów: na liście znajdują się tylko te, które możemy wykonać
* Dodatkowo możemy przeskoczyć do ekranu wszystkich przepisów



# 2.6 Raport odstępstw od wymagań

**Zmiana:** Rezygnacja z wprowadzania systemu miar i jednostek produktów aplikacji

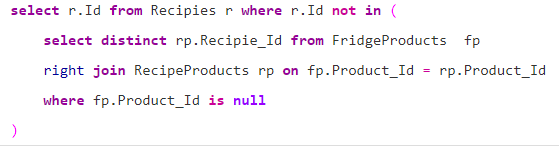
**Uzasadnienie:**

Po zebraniu opinii potencjalnych użytkowników aplikacji ustalono, że niedopracowany system miar i jednostek jest większą przeszkodą niż potrzebną funkcjonalnością. Użytkownik nie zawsze ma czas mierzyć ilość nowo kupionego produktu, jak też często w przepisach ilość danego towaru jest kwestią indywidualną. W dalszych etapach aplikacji rozważane jest wprowadzenie opcjonalnego systemu miar produktów nie wpływającego na wyniki wyszukiwania, a będącego jedynie podpowiedzią.

# 3. Dokumentacja końcowa (powykonawcza) - punkty wymagane przez prowadzącego zajęcia

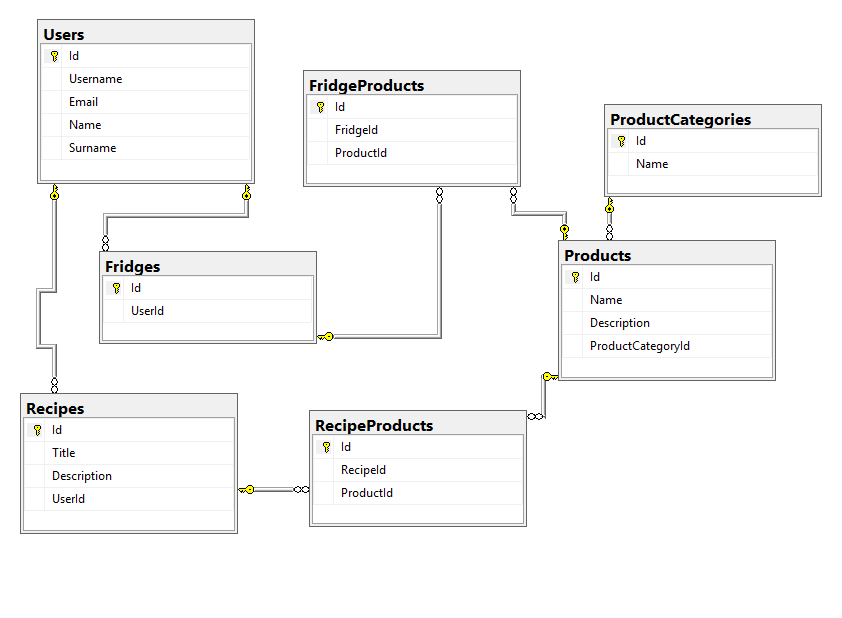
# 3.1 Pseudokod

Kod SQL zapytania, na którym oparliśmy wyrażenie LINQ, które posłużyło nam do szukania przepisów , które można wykonać z produktów w lodówce :



# 3.2 Model danych

Całość bazy danych będzie przechowywana w systemie zarządzania relacyną bazą danych MsSQL. Poniżej opisano zawartość poszczególnych tabel. Skrypt SQL wypełniający bazę danych zawarty jest w projekcie DatabaseSeed , w pliku Seed.sql. Skrypt ten czyści aktualną zawartość bazy i zapełnia przykładowymi danymi. Sam schemat bazy danych został stworzony w myśl podejścia „Code-Frirst” przy użyciu Entity Framework.



Tabele: FridgeProducts i RecipeProducts są tabelami zapewniającymi odpowiednią strukturę bazy danych. Mają na celu zamodelowanie porduktów znajduących się odpowiedni w lodówce i w konkretnych przepisach.

Nazwy kolum i diagram tabel jest wystarczająco samodeskryptywny i nie potrzebuje głębszej analizy.