

Sprawozdanie z projektu laboratoryjnego realizowanego w ramach przedmiotu Programowanie Komputerów IV

Data oddania sprawozdania: 21.06.2020

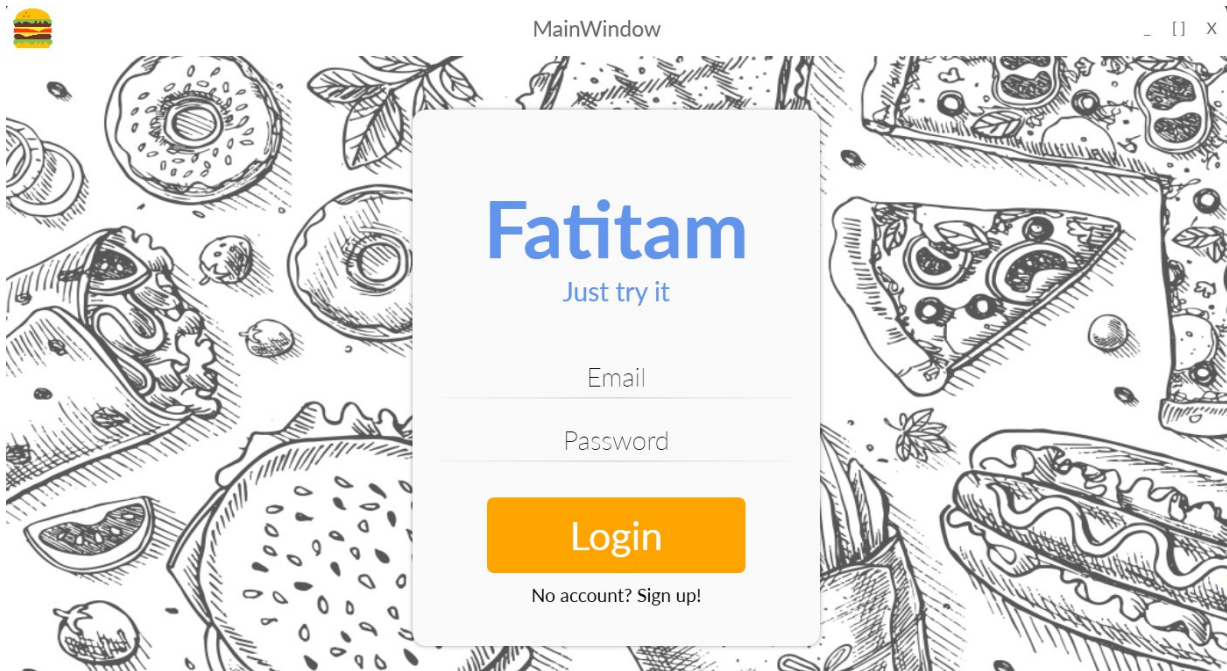
Temat projektu	Aplikacja do liczenia kalorii
Autor	Krzysztof Zgadzaj
Wydział	AEiI
Kierunek	Informatyka
Przedmiot	Programownie Komputerów IV
Prowadzący	mgr inż. Łukasz Dyga

1. Zarys projektu

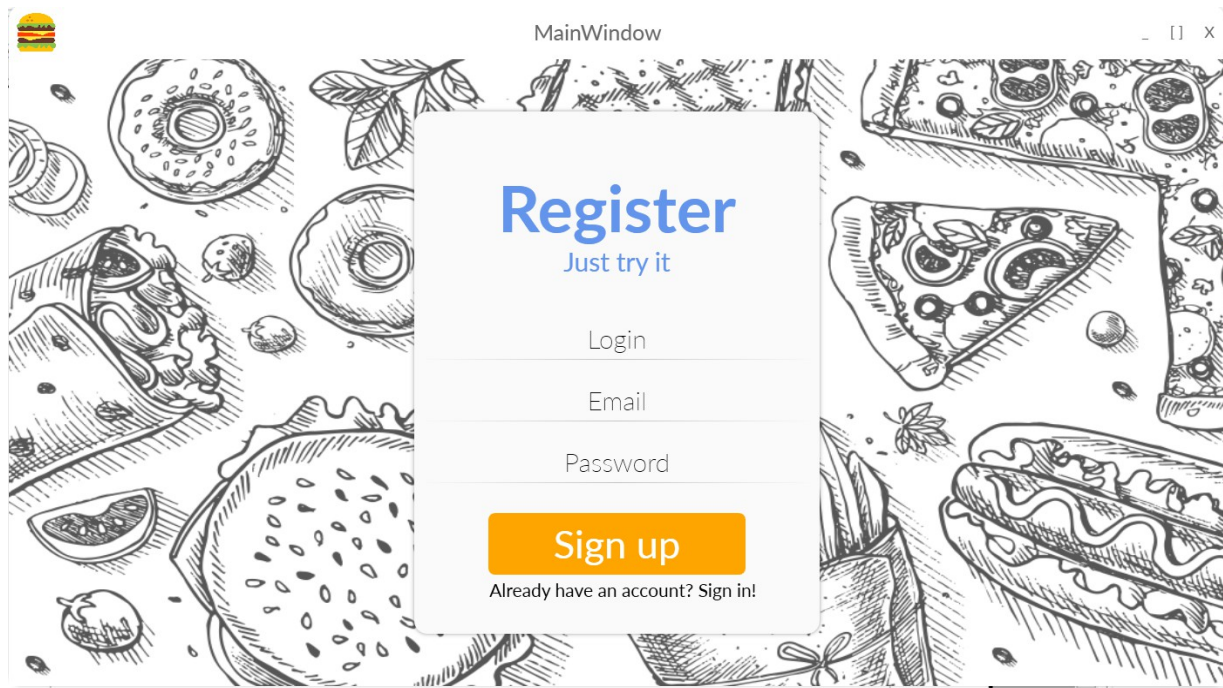
Aplikacja którą stworzyłem umożliwia użytkownikowi liczenie kalorii i makroskładników. W tym celu trzeba najpierw dodać produkt, określając przy tym jego parametry, co umożliwia późniejsze wpisanie go do jadłospisu.

2. Opis działania

Uruchomienie programu spowoduje otwarcie aplikacji na ekranie logowania

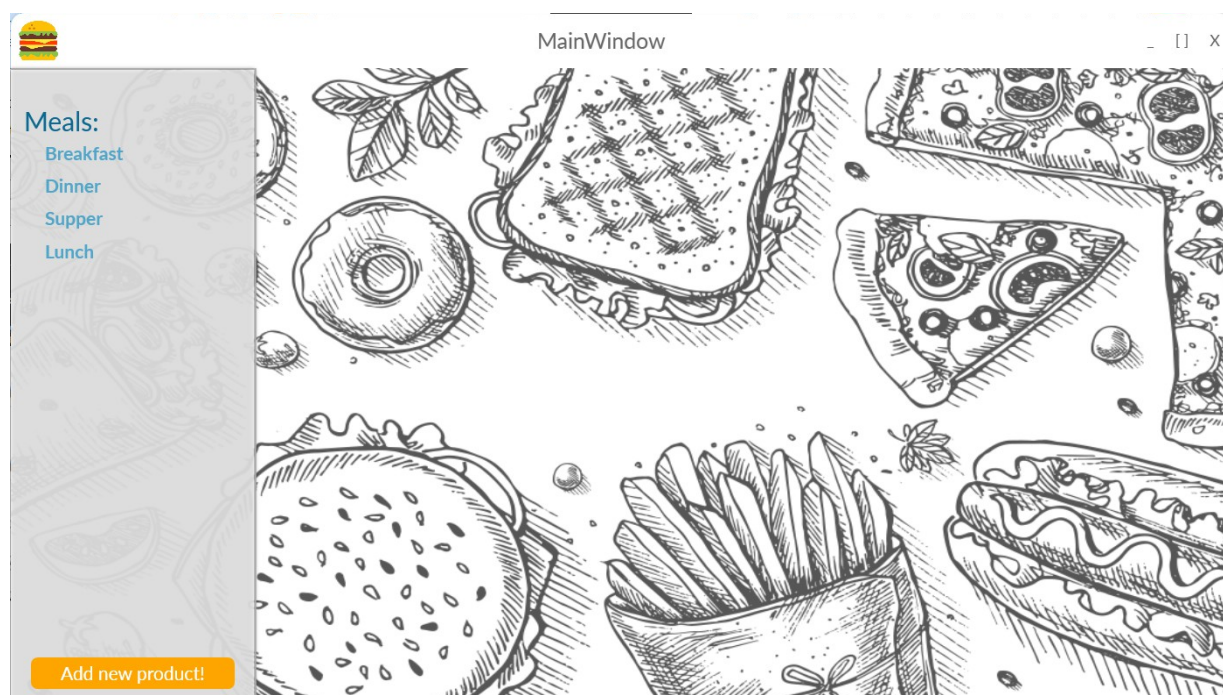


Z tego miejsca możemy przejść do ekranu rejestracji, bądź do głównego okna aplikacji. Poniżej prezentuję pierwszą opcję.

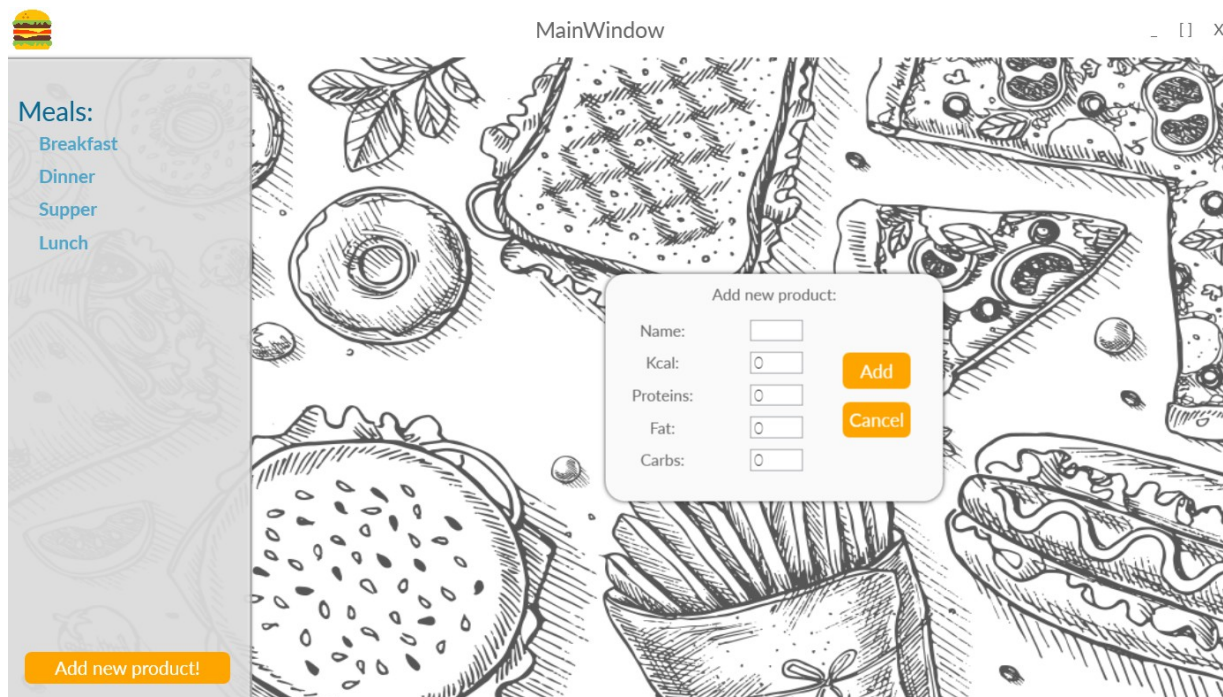


Ze względu na nieprzewidziane problemy implementacyjne zabrakło mi czasu na podłączenie projektu do bazy danych, toteż aplikacja nie jest w pełni funkcjonalna. Gdyby nie to, w bazie danych zapisywane by były dane użytkownika oraz przypisana do konta lista produktów.

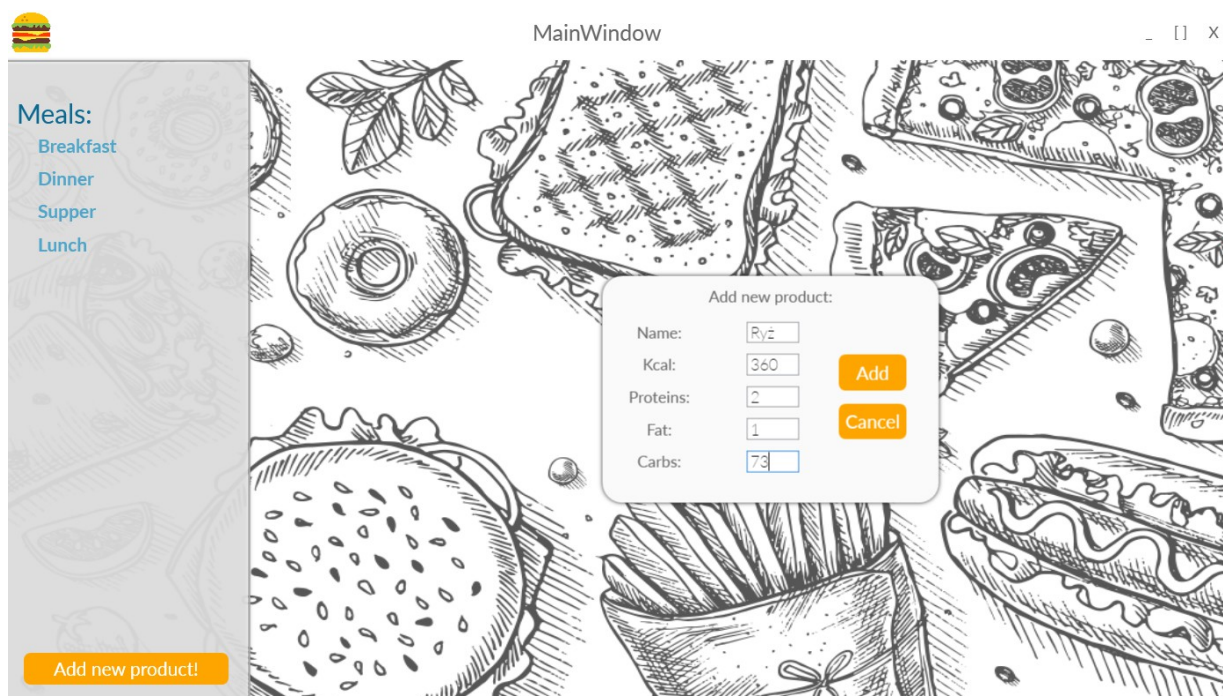
Zaraz po zalogowaniu aplikacja prezentuje się jak poniżej

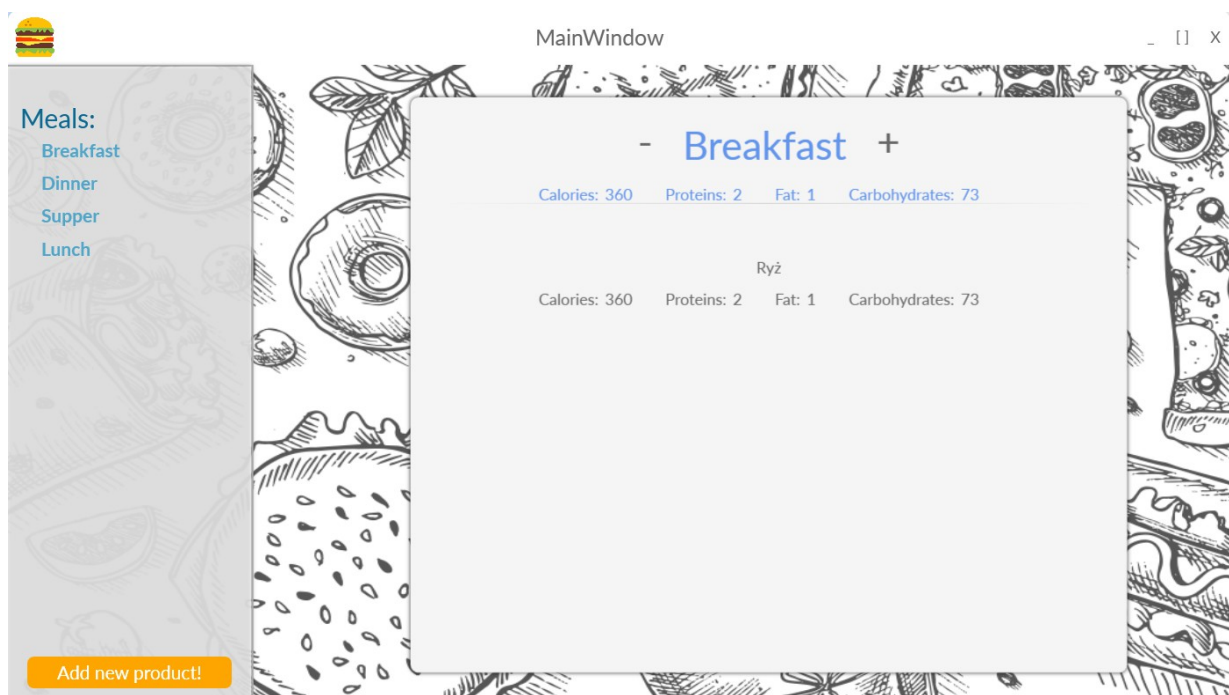
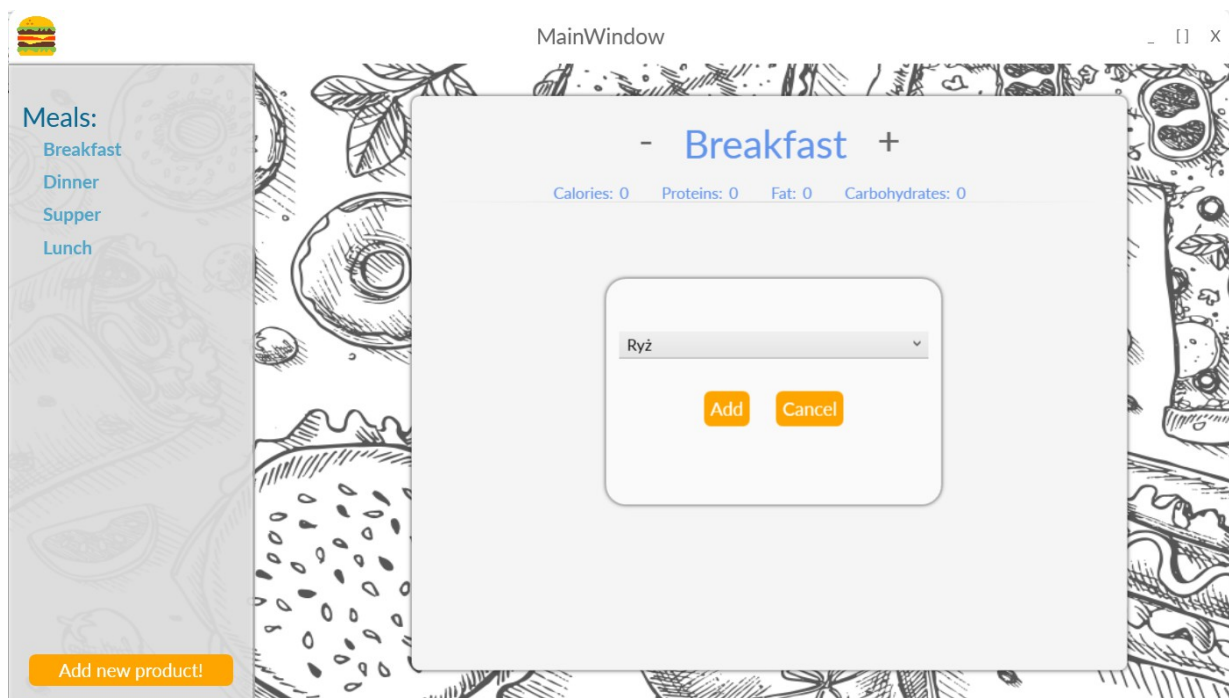


Następnym krokiem powinno być stworzenie produktów



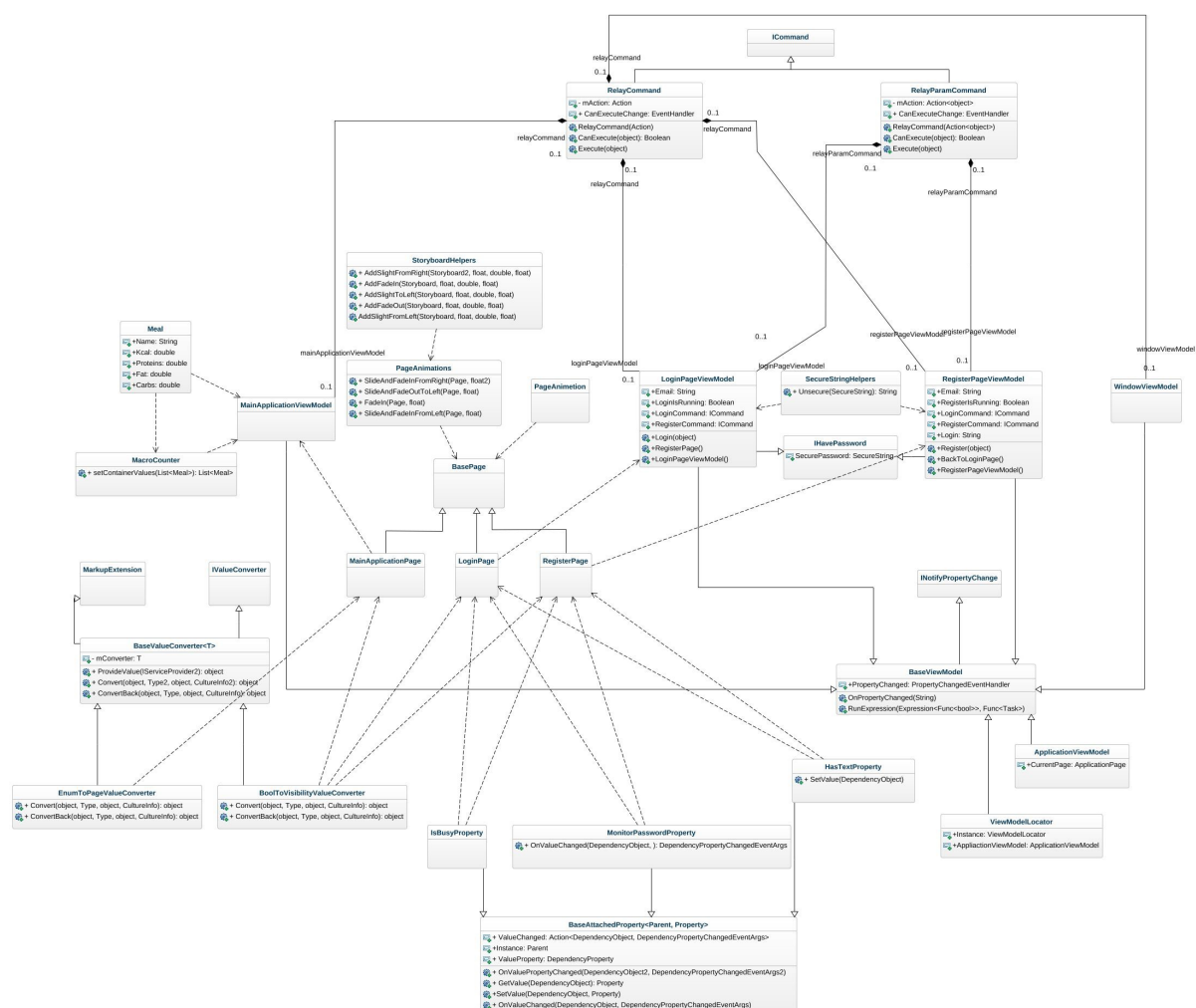
Po wypełnieniu wszystkich pól i wciśnięciu przycisku "Add" produkt zostaje zapisany w liście. Możemy teraz dodać go do dowolnego posiłku





Jak widać, aplikacja działa jak powinna.

3. Diagram klas



4. Laboratorium

W celu realizacji tego projektu, użyłem zagadnień wchodzących w skład następujących ćwiczeń laboratoryjnych:

1) Szablony

```
public abstract class BaseAttachedProperty<Parent, Property>
    where Parent : BaseAttachedProperty<Parent, Property>, new()
{
```

```
public abstract class BaseValueConverter<T> : MarkupExtension, IValueConverter
{
    where T : class, new()
```

2) Kontenery STL

W projekcie zostało użyte wiele kontenerów, takich jak List czy IKernel

```
public static IKernel Kernel { get; private set; } = new StandardKernel();
```

```
public List<Meal> BreakfastList { get; set; } = new List<Meal>();  
public List<Meal> DinnerList { get; set; } = new List<Meal>();  
public List<Meal> SupperList { get; set; } = new List<Meal>();  
public List<Meal> LunchList { get; set; } = new List<Meal>();
```

3) Algorytmy i iteratory STL

Na każdym etapie pisania projektu korzystałem z interfejsu IEnumerable. Dodatkowo użyta została instrukcja foreach.

```
public static Meal setContainerValues(List<Meal> meals)  
{  
    Meal temp = new Meal { Name = "", Kcal = 0, Fat = 0, Carbs = 0, Proteins = 0 };  
  
    foreach (Meal m in meals)  
    {  
        temp.Kcal += m.Kcal;  
        temp.Proteins += m.Proteins;  
        temp.Carbs += m.Carbs;  
        temp.Fat += m.Fat;  
    }  
  
    return temp;  
}
```

4) Inteligentne wskaźniki

W C# wskaźniki domyślnie są inteligentne. Korzystam z takowych tworząc np. Obiekty typu ICommand.

```
public ICommand LoginCommand { get; set; }  
public ICommand GoToRegisterCommand { get; set; }
```

5. Trudności podczas wykonywania projektu i podsumowanie.

Opisany powyżej program jest moim pierwszym napisanym w C#. W związku z tym, wykonanie niektórych operacji, bądź zrozumienie kodu znalezionej na różnych stronach internetowych zajęło mi znacznie więcej czasu niż gdybym pisał w języku C++. Problemy sprawiło mi też programowanie w XAML-u, gdyż nigdy wcześniej nie spotkałem się z tego typu składnią. W efekcie nie zdążyłem zaimplementować całej funkcjonalności. Jak już pisałem wcześniej, nie podłączyłem programu do bazy danych. Dodatkowo, w kodzie jest kilka błędów, których nie udało mi się poprawić. Przykładowo, po dodaniu elementu do któregoś z dań, niemożliwe jest dodanie kolejnego produktu do listy wszystkich produktów. Mimo tych niedoskonałości, uważam, że bardzo dużo się nauczyłem w trakcie pisania tego projektu i na pewno będę kontynuował pracę z tą technologią.