



Praca dyplomowa – inżynierska

**PROJEKT I IMPLEMENTACJA  
SYSTEMU DO ZARZĄDZANIA DIETĄ  
W OPARCIU O ARCHITEKTURĘ MIKROSERWISÓW**

Krzysztof Marczyński

słowa kluczowe:

dieta, jadłospisy, aplikacja webowa, mikroserwisy

krótkie streszczenie:

W pracy przedstawiono projekt aplikacji służącej do układania diet.

Opiekun pracy dyplomowej	dr inż. Michał Szczepanik	.....	.....
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego		.....	.....
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis

Do celów archiwalnych pracę dyplomową zakwalifikowano do:\*

- a) kategorii A (akta wieczyste)
- b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

\* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław, rok 2019



## Streszczenie

1.Celem pracy było opracowanie aplikacji służącej do komunikacji z kosmitami. Dostępne na rynku aplikacje nie satysfakcjonowały autorki ze względu na brak istotnych funkcji takich jak obsługa przez telefon z systemem Android. W ramach pracy przygotowano aplikację komunikacyjną wykorzystującą framework SpaceDirect, przechowującą dane kontaktów w bazie danych MyNoSQL oraz udostępniającą swoje funkcje przez interfejs REST API. Oprócz projektu aplikacji praca zawiera wyniki testów jednostkowych oraz testów użyteczności przeprowadzonych przez krewnych i znajomych królika. Przygotowana w ramach projektu inżynierskiego praca może zostać wykorzystana przez wszystkie osoby zainteresowane kontaktami z cywilizacjami pozaziemskimi.

## Abstract

2.The main goal of this thesis was development of...(please translate remaining part of Streszczenie into English).



# Spis treści

<b>Wstęp</b>	<b>2</b>
Opis problemu . . . . .	2
Cel pracy . . . . .	2
Zakres pracy . . . . .	2
<b>1. Wymagania projektowe</b>	<b>3</b>
1.1. Sformułowanie problemu . . . . .	3
1.2. Pozycjonowanie produktu . . . . .	3
1.3. Opis udziałowców i użytkowników . . . . .	3
1.3.1. Podsumowanie udziałowców . . . . .	3
1.3.2. Podsumowanie użytkowników . . . . .	3
1.4. Słownik . . . . .	3
<b>2. Stan wiedzy i techniki w zakresie tematyki pracy</b>	<b>5</b>
2.1. Przegląd istniejących rozwiązań konkurencyjnych . . . . .	5
2.2. Przegląd przydatnych technologii i technik . . . . .	5
<b>3. Założenia projektowe</b>	<b>6</b>
3.1. Wymagania funkcjonalne . . . . .	6
3.2. Wymagania нефunkcjonalne . . . . .	9
3.3. Opis podstawowej architektury systemu . . . . .	9
<b>4. Projekt</b>	<b>10</b>
4.1. Przypadki użycia . . . . .	10
4.2. Kategorie . . . . .	10
4.3. Reguły funkcjonowania . . . . .	10
4.3.1. Produkty . . . . .	10
4.3.2. Przepisy . . . . .	10
4.3.3. Jadłospisy . . . . .	10
4.4. Ograniczenia dziedzinowe . . . . .	10
4.5. Model domenowy . . . . .	10
4.6. Prototyp interfejsu . . . . .	13
4.7. Baza danych . . . . .	21
<b>5. Implementacja</b>	<b>22</b>
5.1. Wykorzystywane środowiska i narzędzia programistyczne . . . . .	22
5.2. Architektura systemu . . . . .	22
5.3. Instalacja oprogramowania . . . . .	22
5.3.1. Wymagania wstępne . . . . .	22
5.4. Instrukcja użytkowania . . . . .	22

<b>6. Testy</b>	<b>23</b>
6.1. Testy jednostkowe . . . . .	23
6.2. Testy integracyjne . . . . .	23
6.3. Testy akceptacyjne . . . . .	23
<b>Zakończenie</b>	<b>24</b>
6.4. Podsumowanie pracy . . . . .	24
6.5. Możliwości dalszego rozwoju . . . . .	24
<b>Bibliografia</b>	<b>25</b>
<b>Spis kodów źródłowych</b>	<b>27</b>
<b>Dodatek A. To powinien być dodatek</b>	<b>28</b>

3.uzupełnić streszczenie i słowa kluczowe na stronie tytułowej

## Wstęp

4.W dzisiejszym świecie wykorzystanie aplikacji do kontaktów z kosmitami wydaje się oczywiste. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## Opis problemu

5.O ile sposób komunikacji nie budzi już dziś wątpliwości, to dostępne na rynku aplikacje nie zaspokajają potrzeb wymagającego użytkownika, ograniczając możliwości np. zmiany kolorystyki interfejsu użytkownika z jasnej na ciemną. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

## Cel pracy

6.Celem pracy jest projekt i budowa platformy do zarządzania dietą w oparciu o architekturę mikroserwisów. Tworzona platforma będzie obejmowała cały cykl życia diety, czyli przede wszystkim: zebranie przez dietetyka wywiadu żywieniowego od pacjenta, stworzenie przez dietetyka jadłospisu, udostępnienie jadłospisu pacjentowi i elementy pomagające pacjentowi stosować dietę. Praca w swoim zakresie będzie zawierała, między innymi, współpracę z ekspertami domenowymi w dziedzinie dietetyki, projekt platformy oraz jej realizację z wykorzystaniem wybranych narzędzi programistycznych.

## Zakres pracy

7.Praca obejmowała opracowanie projektu aplikacji, implementację w języku JavaScript oraz wdrożenie wszystkiego powyższego na platformie GitHub. Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.



## 1. Wymagania projektowe

### 1.1. Sformułowanie problemu

8.sformułowanie problemu

### 1.2. Pozycjonowanie produktu

9.pozycjonowanie produktu

### 1.3. Opis udziałowców i użytkowników

#### 1.3.1. Podsumowanie udziałowców

10.opis udziałowców

#### 1.3.2. Podsumowanie użytkowników

11.poprawić użytkowników

Tabela 1.1: Użytkownicy (opr.wł)

Nazwa	Opis	Odpowiedzialności
Administrator	Osoba zarządzająca działaniem aplikacji	<ul style="list-style-type: none"><li>• Przydzielanie i odbieranie użytkownikom uprawnień</li></ul>
Dietetyk	Specjalista w dziedzinie dietetyki	<ul style="list-style-type: none"><li>• Używa założonego konta</li><li>• Wprowadza, edytuje i usuwa produkty, przepisy i jadłospisy</li></ul>
Pacjent	Klient dietetyka	<ul style="list-style-type: none"><li>• Otrzymuje (mailowo) ułożony jadłospis</li></ul>

### 1.4. Słownik

12.uzupełnić słownik

- Administrator – użytkownik posiadający uprawnienia do zarządzania uprawnieniami użytkowników
- Dietetyk – specjalista w dziedzinie dietetyki

- Jadłospis – plan posiłków zdefiniowany na określoną liczbę dni z uwzględnieniem określonych wymagań
- Karta pacjenta – karta przedstawiająca przebieg współpracy dietetyka z pacjentem
- Miara domowa – definicja pospolitej miary, takiej jak np. łyżeczka w gramach
- Pacjent – klient dietetyka
- Pomiary ciała – pomiary ciała pacjenta przeprowadzane przez dietetyka
- Posiłek – posiłek jest przydzielany do jadłospisu; zawiera produkty i przepisy
- Produkt – produkt spożywczy, dla którego specyfikowane są wartości odżywcze i miary domowe
- Przepis – opis składników i kroków przygotowania dania
- Sekcja przepisu – semantyczny podział przepisu, np. sernik może mieć sekcje związane z przygotowaniem ciasta, nadzienia i polewy
- Wartość odżywcza – ilość elementu takiego jak np. węglowodanów albo białka w 100g produktu
- Wizyta – konkretna wizyta pacjenta
- Wywiad żywieniowy – wywiad przeprowadzany z pacjentem uwzględniający jego nawyki żywieniowe, nietolerancje, choroby, przyjmowane leki, itp.

## **2. Stan wiedzy i techniki w zakresie tematyki pracy**

### **2.1. Przegląd istniejących rozwiązań konkurencyjnych**

13.opisać rozwiązania konkurencyjne

- Dietico
- TiqDiet
- Kcalmar PRO
- Dietetyk Pro

### **2.2. Przegląd przydatnych technologii i technik**

14.Tutaj opisać architekturę aplikacji webowych. Porównać monolit, soa i mikroserwisy

### 3. Założenia projektowe

#### 3.1. Wymagania funkcjonalne

15.uzupełnić wymagania funkcjonalne

Tabela 3.1: Wymagania funkcjonalne ogólne (opr.wł)

Potrzeby	Cechy
Administrator potrzebuje widzieć listę użytkowników	<ul style="list-style-type: none"><li>• Przydzielanie i odbieranie użytkownikom uprawnień</li></ul>

Tabela 3.2: Wymagania funkcjonalne dla produktów (opr.wł)

Potrzeby	Cechy
Administrator potrzebuje zarządzać definicjami potrzebnymi w produktach	<ul style="list-style-type: none"><li>• Zarządzanie definicjami wartości odżywczych</li><li>• Zarządzanie kategoriami produktów</li><li>• Zarządzanie rodzajami diet</li></ul>
Dietetyk potrzebuje widzieć listę produktów	<ul style="list-style-type: none"><li>• Wyszukiwanie produktów</li><li>• Filtrowanie produktów</li><li>• Dodawanie nowych produktów</li></ul>
Dietetyk potrzebuje zarządzać szczegółami produktu	<ul style="list-style-type: none"><li>• Edytowanie i usuwanie produktów</li><li>• Definiowanie wartości odżywczych dla produktu</li><li>• Definiowanie miar domowych dla produktu</li><li>• Przypisywanie produktu do kategorii i podkategorii</li><li>• Definiowanie do jakich typów diet produkt nadaje się a do jakich nie</li></ul>

Tabela 3.3: Wymagania funkcjonalne dla przepisów (opr.wł)

Potrzeby	Cechy
Administrator potrzebuje zarządzać definicjami potrzebnymi w przepisach	<ul style="list-style-type: none"> <li>• Zarządzanie typami posiłków</li> <li>• Zarządzanie typami dań</li> <li>• Zarządzanie definicjami wyposażenia kuchennego</li> </ul>
Dietetyk potrzebuje widzieć listę przepisów	<ul style="list-style-type: none"> <li>• Wyszukiwanie przepisów</li> <li>• Filtrowanie przepisów</li> <li>• Dodawanie nowych przepisów</li> </ul>
Dietetyk potrzebuje zarządzać szczegółami przepisu	<ul style="list-style-type: none"> <li>• Edytowanie i usuwanie przepisów</li> <li>• Dodawanie wielu sekcji do przepisu</li> <li>• Dodawanie do każdej sekcji listy składników</li> <li>• Dodawanie do każdej sekcji sposobu przygotowania</li> <li>• Dodawanie zdjęcia dania do przepisu</li> <li>• Definiowanie czasu przygotowania posiłku</li> </ul>

Tabela 3.4: Wymagania funkcjonalne dla jadłospisów (opr.wł)

Potrzeby	Cechy
Dietetyk potrzebuje widzieć listę jadłospisów	<ul style="list-style-type: none"> <li>• Wyszukiwanie jadłospisów</li> <li>• Filtrowanie jadłospisów</li> <li>• Dodawanie nowych jadłospisów</li> </ul>
Dietetyk potrzebuje zarządzać szczegółami jadłospisu	<ul style="list-style-type: none"> <li>• Dodawanie, edytowanie i usuwanie jadłospisów</li> <li>• Definiowanie liczby dni na które będzie układany jadłospis</li> <li>• Definiowanie liczby posiłków dziennie</li> <li>• Definiowanie planowanego czasu każdego z posiłków</li> <li>• Definiowanie procentowego udziału podstawowych wartości odżywczych w każdym posiłku</li> <li>• Definiowanie posiłków w jadłospisie</li> <li>• Dodawanie produktów i przepisów do posiłków</li> </ul>

Tabela 3.5: Wymagania funkcjonalne dla wizyt (opr.wł)

Potrzeby	Cechy
Dietetyk potrzebuje wyświetlać listę swoich pacjentów	<ul style="list-style-type: none"> <li>• Wyszukiwanie pacjentów</li> <li>• Wyświetlanie listy znalezionych pacjentów</li> <li>• Wyświetlanie listy umówionych wizyt</li> <li>• Wyświetlanie listy oczekujących porad</li> <li>• Dodawanie nowych pacjentów</li> </ul>
Dietetyk potrzebuje zarządzać kartą pacjenta	<ul style="list-style-type: none"> <li>• Wyświetlanie i edytowanie podstawowych informacji pacjenta</li> <li>• Wyświetlanie listy wizyt pacjenta</li> <li>• Wyświetlanie listy oczekujących porad pacjenta</li> <li>• Dodawanie nowej wizyty pacjenta</li> </ul>
Dietetyk potrzebuje wyświetlać szczegóły wizyty pacjenta	<ul style="list-style-type: none"> <li>• Wyświetlanie i edytowanie szczegółów wizyty pacjenta</li> <li>• Zarządzanie pomiarami ciała pacjenta przypisanymi do wizyty</li> <li>• Zarządzanie wywiadem żywieniowym przypisanym do wizyty</li> <li>• Zarządzanie jadłospisem przydzielonym do wizyty</li> </ul>
Pacjent potrzebuje otrzymywać dietę	<ul style="list-style-type: none"> <li>• Automatyczne wysyłanie do pacjenta diety mailem po zatwierdzeniu jadłospisu przez dietetyka</li> </ul>
Pacjent chce mieć wgląd w swoją kartę	<ul style="list-style-type: none"> <li>• Logowanie do konta utworzonego w serwisie</li> <li>• Dodawanie kart pacjenta do swojego konta po udostępnieniu ich przez dietetyka</li> </ul>
Pacjent zarządza dietą za pomocą asystenta głosowego	<ul style="list-style-type: none"> <li>• Wydawanie w języku naturalnym poleceń dotyczących diety</li> </ul>

### 3.2. Wymagania niefunkcjonalne

#### 16.uzupełnić wymagania niefunkcjonalne

- System działa poprawnie w przeglądarce Google Chrome i Firefox
- System działa poprawnie w przeglądarce Edge i Opera
- System działa poprawnie w przeglądarce Internet Explorer i Safari
- System działa na urządzeniach mobilnych Android
- System działa na urządzeniach mobilnych iOS
- System jest dostępny w polskiej wersji językowej
- System ma czytelny i ładny interfejs
- Aplikacja webowa jest w pełni responsywna i wygodna do używania niezależnie od wielkości ekranu urządzenia klienckiego
- Aplikacja webowa udostępnia część funkcji offline
- Aplikacja ma być oparta na architekturze mikroserwisów

### 3.3. Opis podstawowej architektury systemu

#### 17.Opisać, że to aplikacja webowa w architekturze mikroserwisów Wyszczególnienie modułów; Diagram rozmieszczenia

## **4. Projekt**

### **4.1. Przypadki użycia**

18.diagram przypadków użycia

### **4.2. Kategorie**

19.uzupełnić kategorie

**KAT/001** Product

**KAT/002** Language

**KAT/003** test22

**KAT/004** test3

### **4.3. Reguły funkcjonowania**

20.uzupełnić reguły funkcjonowania

#### **4.3.1. Produkty**

**REG/001** test1

**REG/002** test2

#### **4.3.2. Przepisy**

**KAT/001** Product

#### **4.3.3. Jadłospisy**

**REG/003** test22

**REG/004** test3

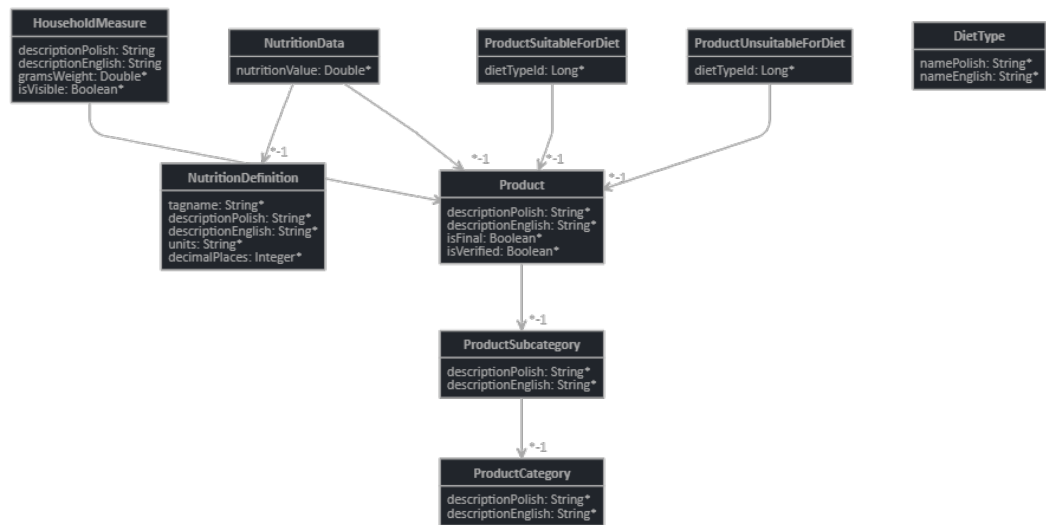
### **4.4. Ograniczenia dziedziczne**

21.uzupełnić ograniczenia dziedziczne

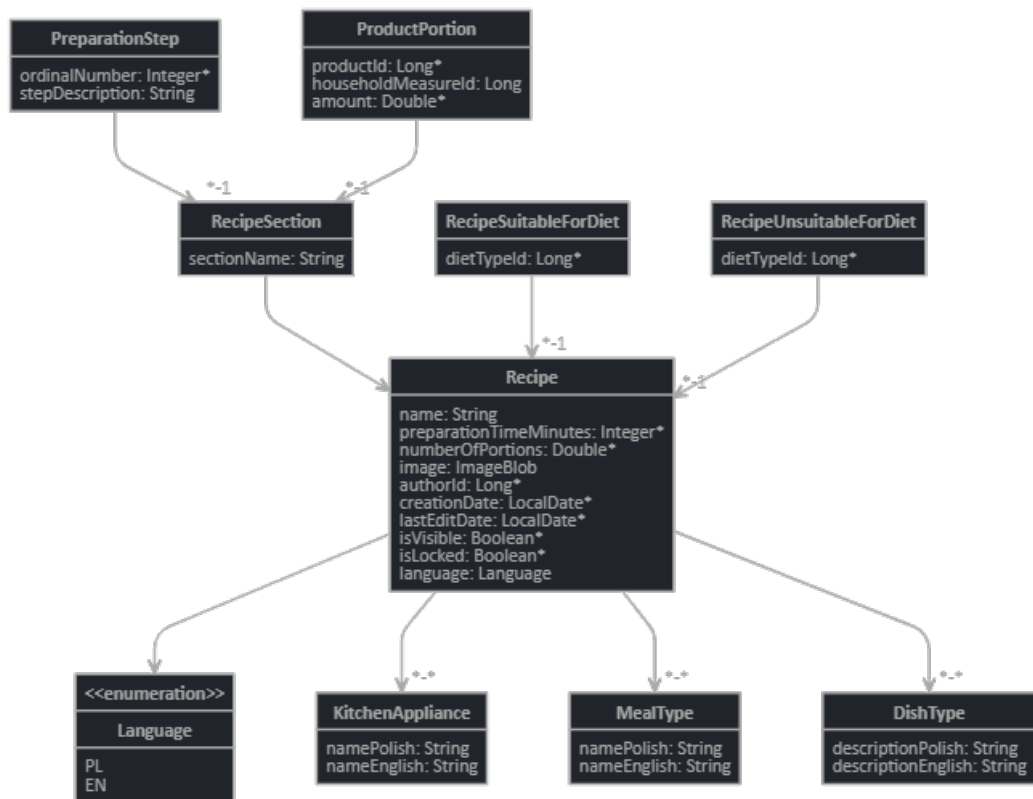
### **4.5. Model domenowy**

22.diagram klas

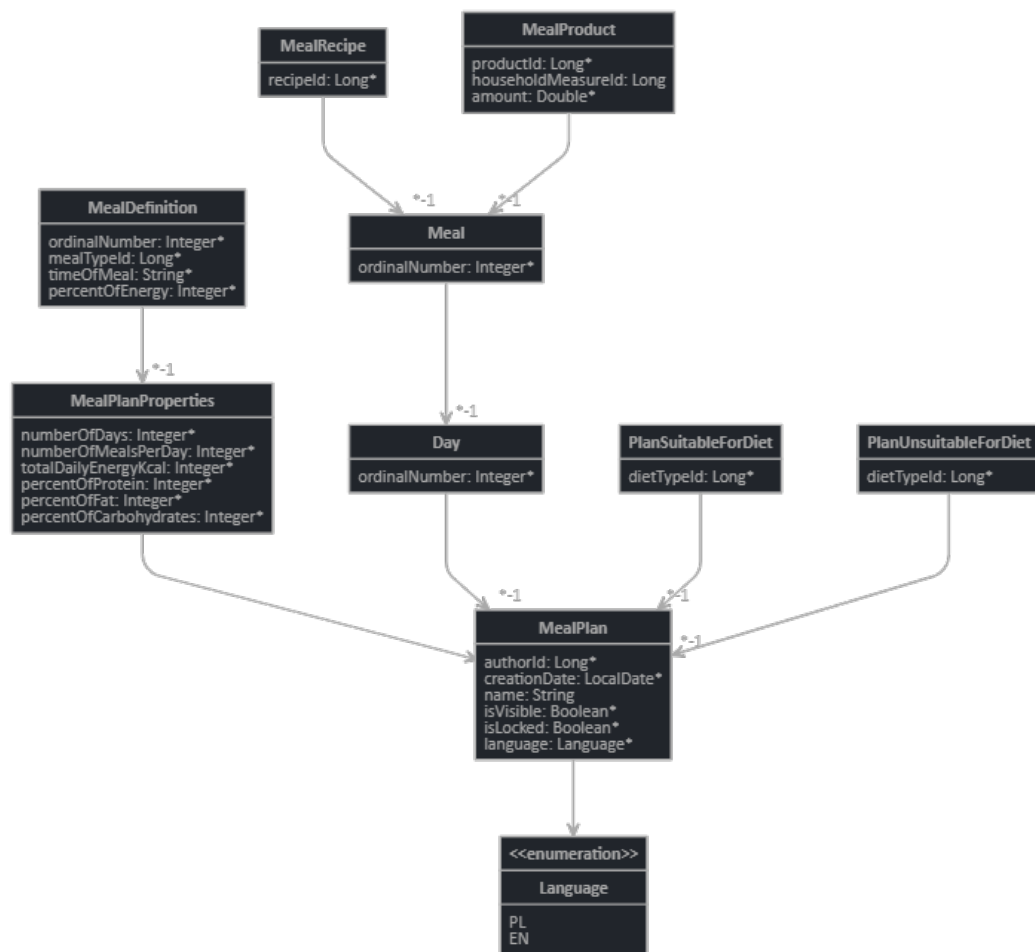




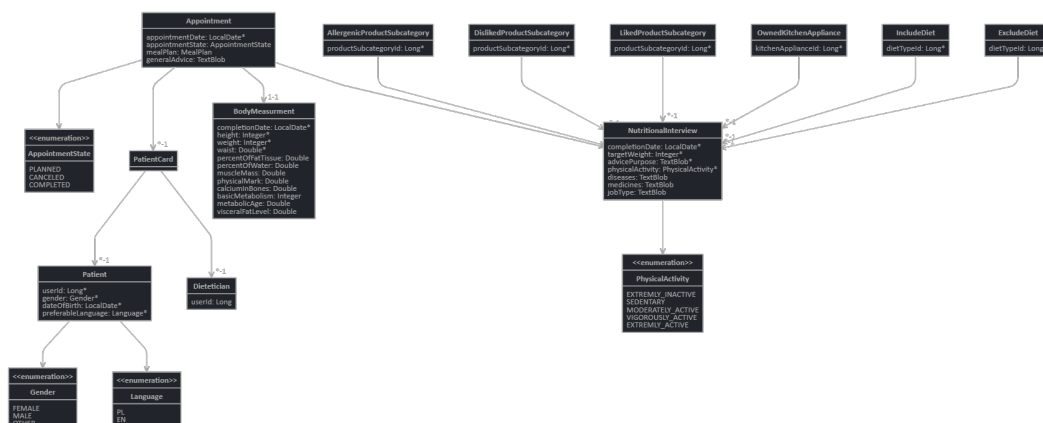
Rys. 4.1: Produkty (opr.wł).



Rys. 4.2: Przepisy (opr.wł).



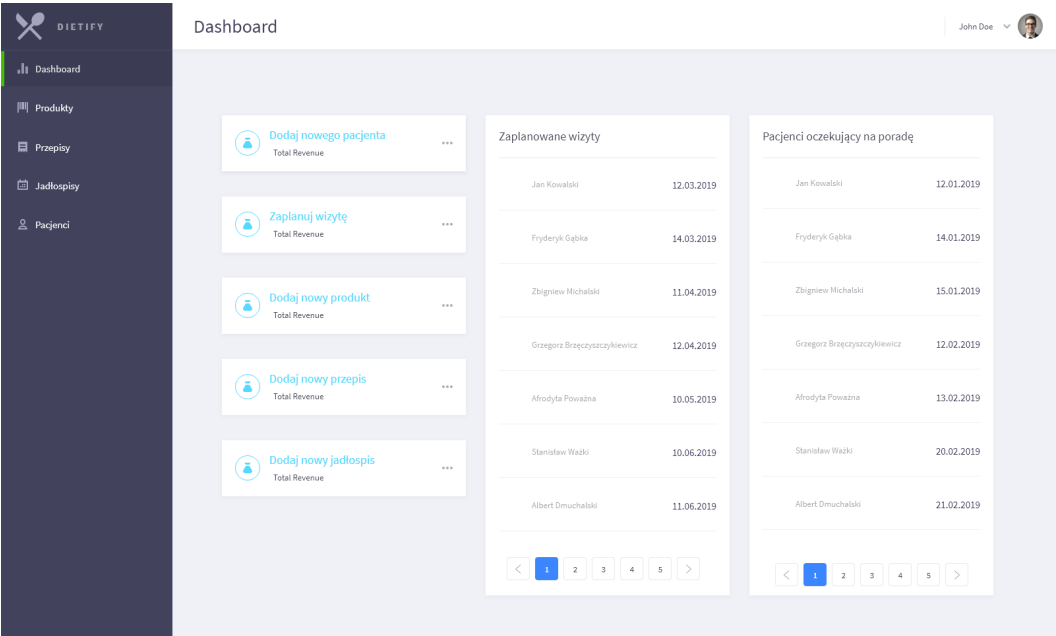
Rys. 4.3: Jadłospisy (opr.wł).



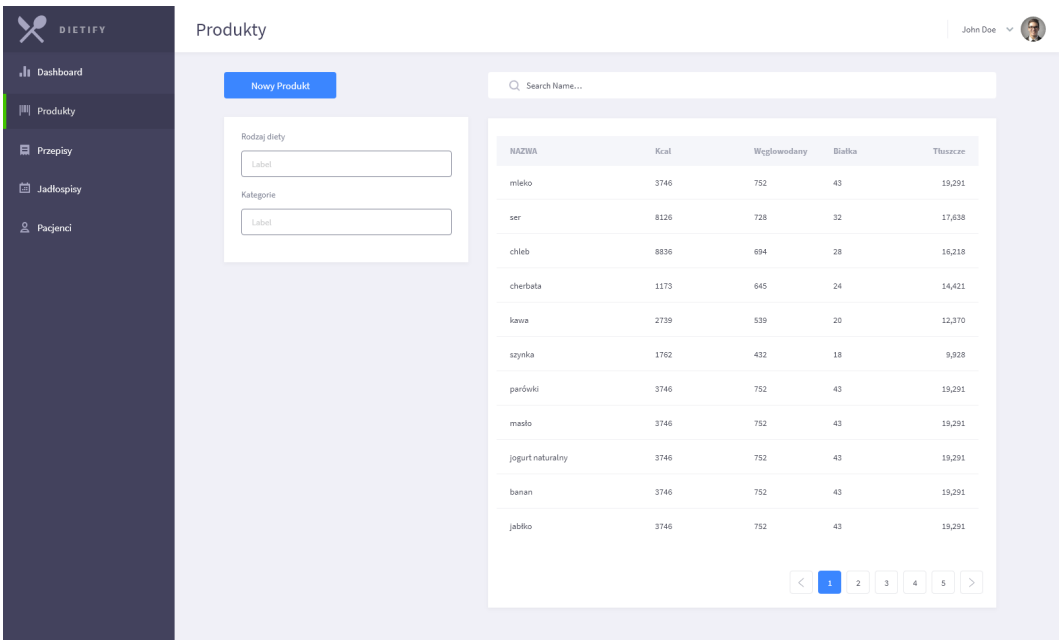
Rys. 4.4: Wizyty (opr.wł).

## 4.6. Prototyp interfejsu

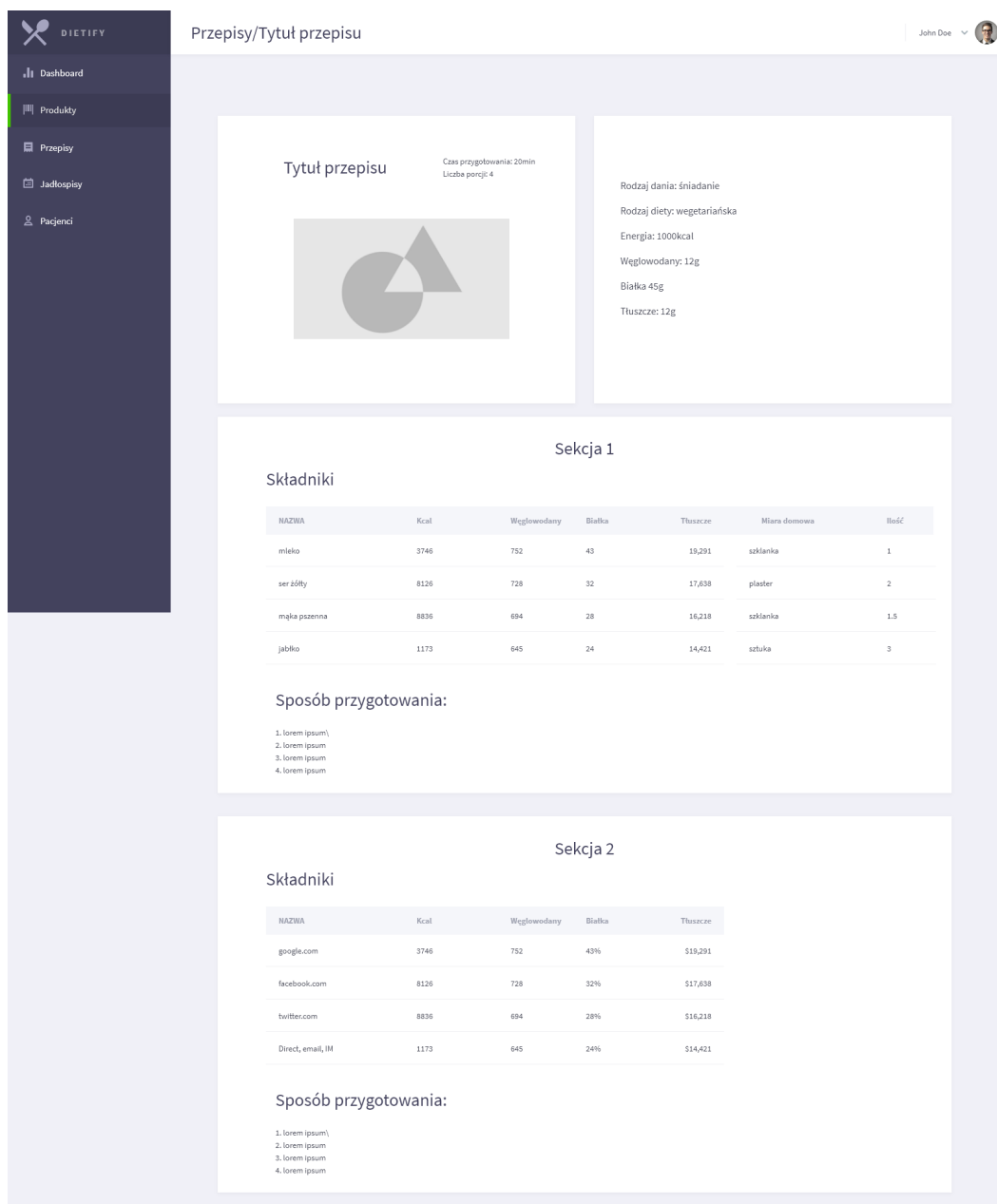
23.mockupy



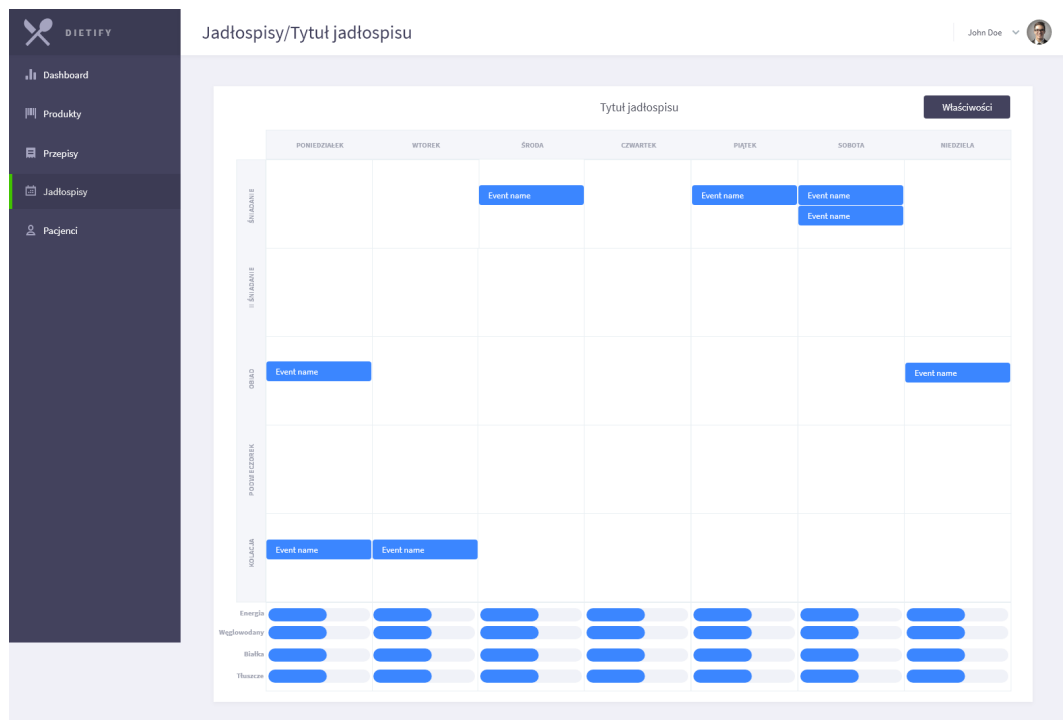
Rys. 4.5: Mockup1 (opr.wł).



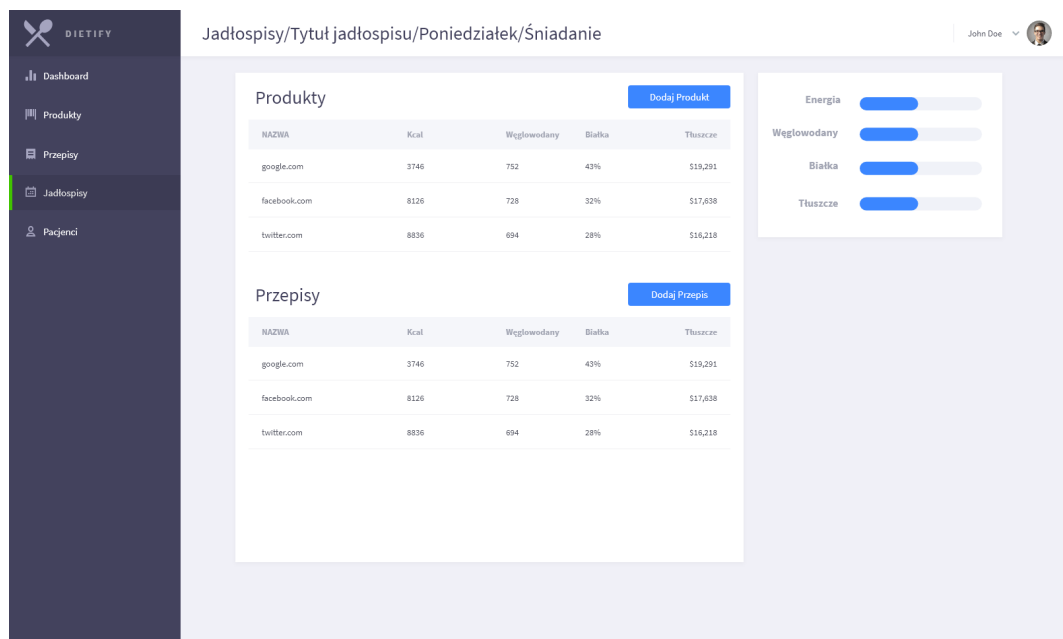
Rys. 4.6: Mockup2 (opr.wł).



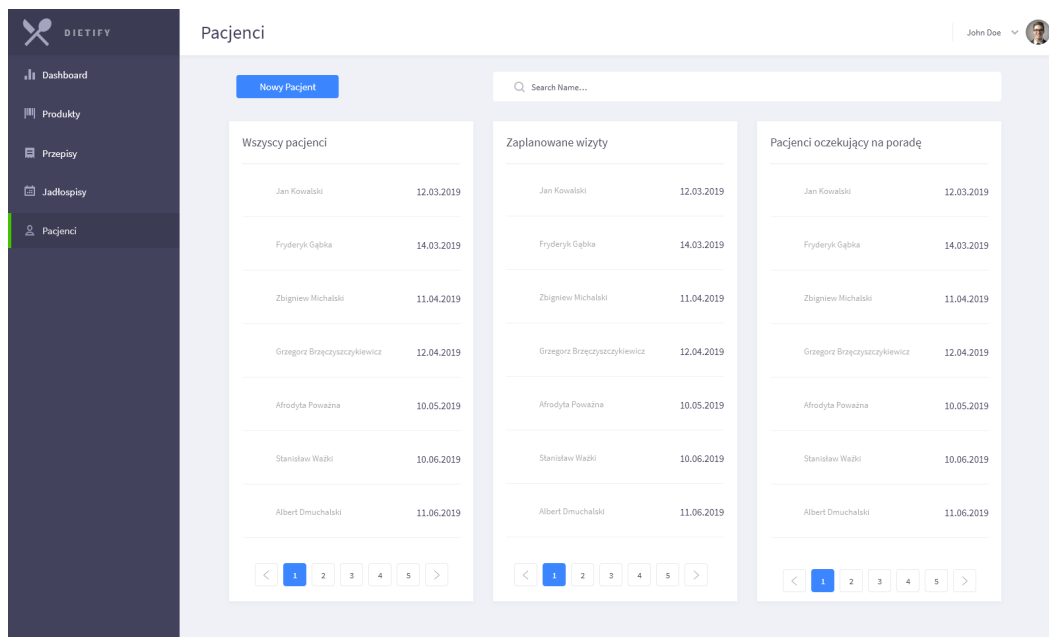
Rys. 4.7: Mockup3 (opr.wł).



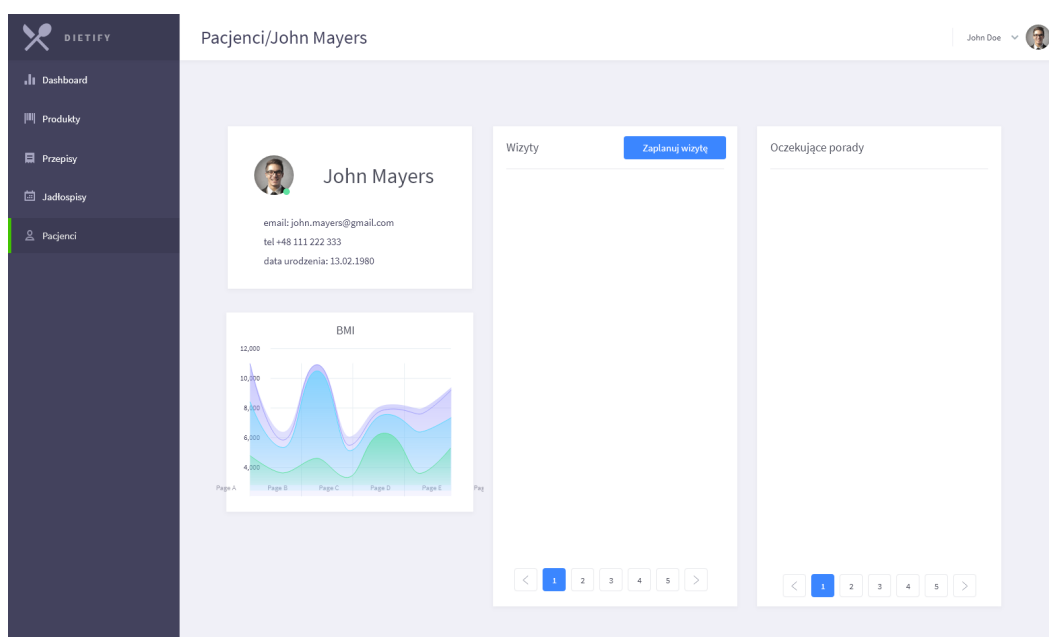
Rys. 4.8: Mockup4 (opr.wł).



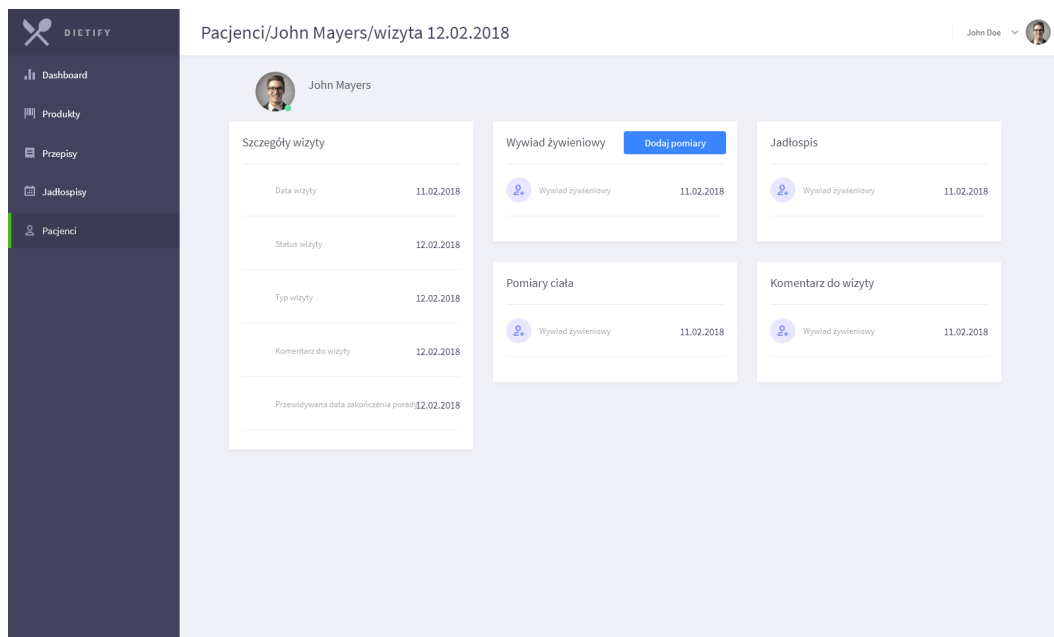
Rys. 4.9: Mockup5 (opr.wł).



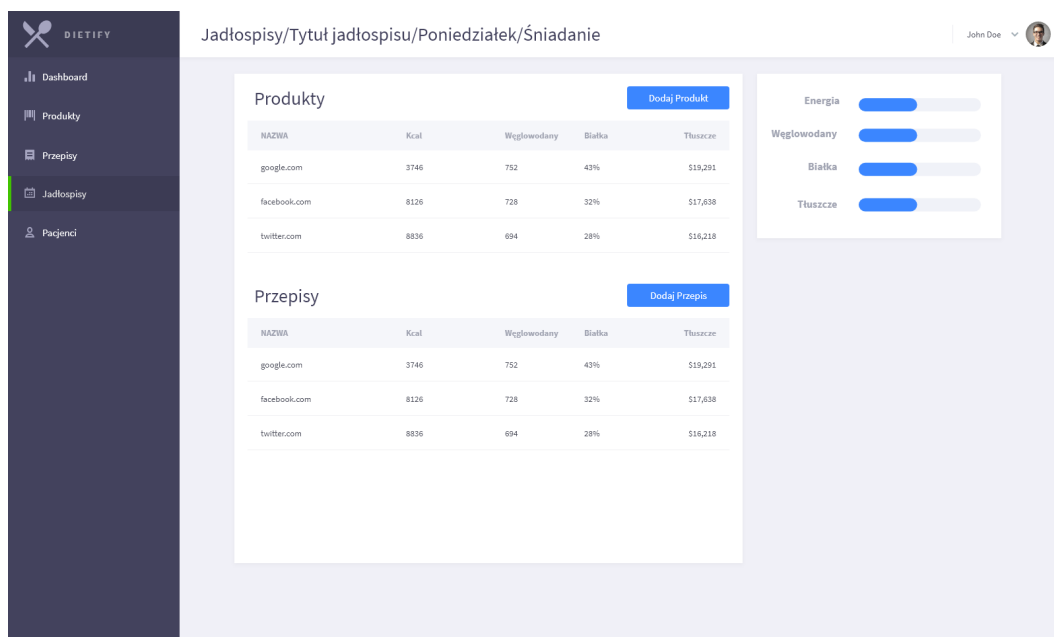
Rys. 4.10: Mockup6 (opr.wł).



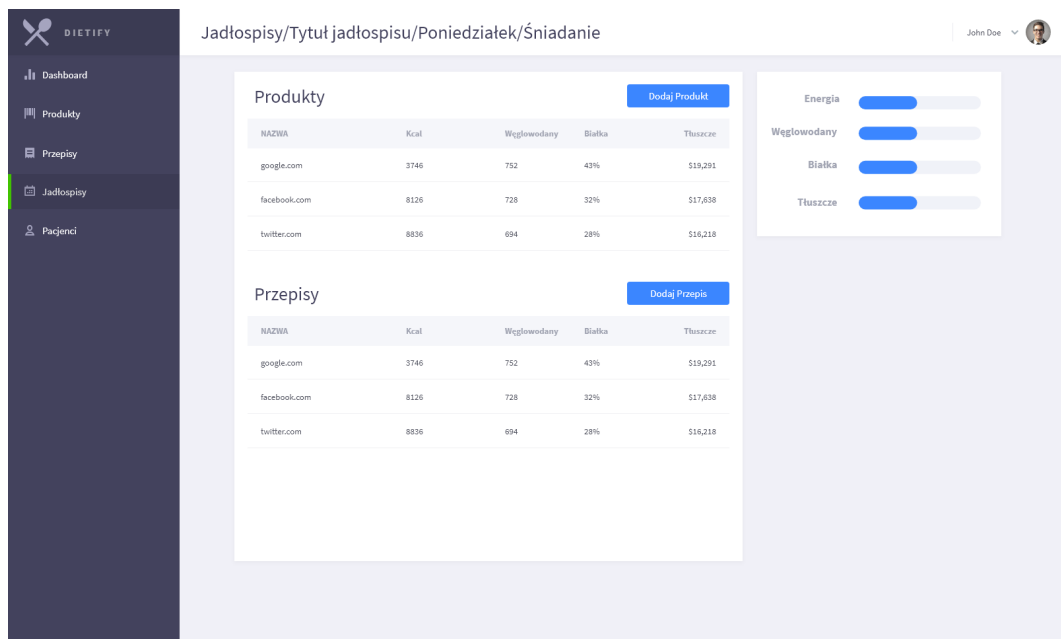
Rys. 4.11: Mockup7 (opr.wł).



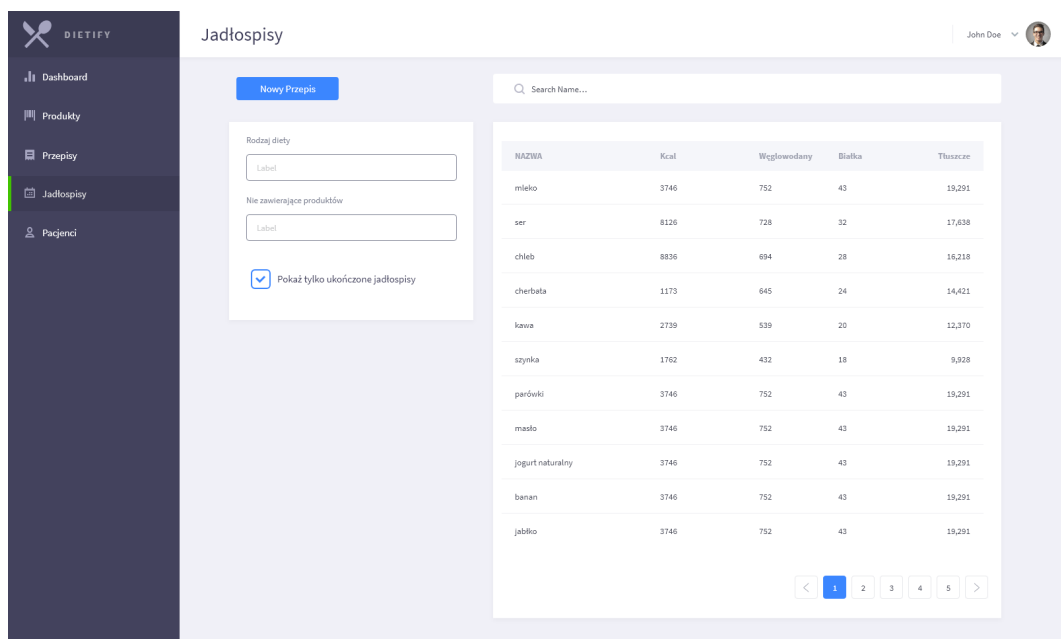
Rys. 4.12: Mockup8 (opr.wł.).



Rys. 4.13: Mockup9 (opr.wł.).

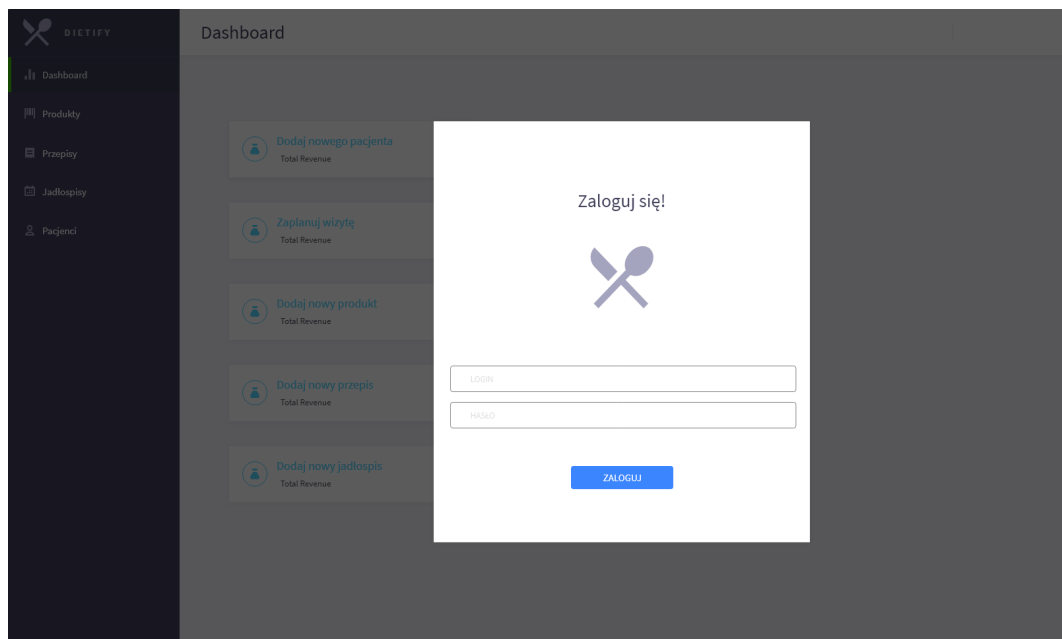


Rys. 4.14: Mockup10 (opr.wł).

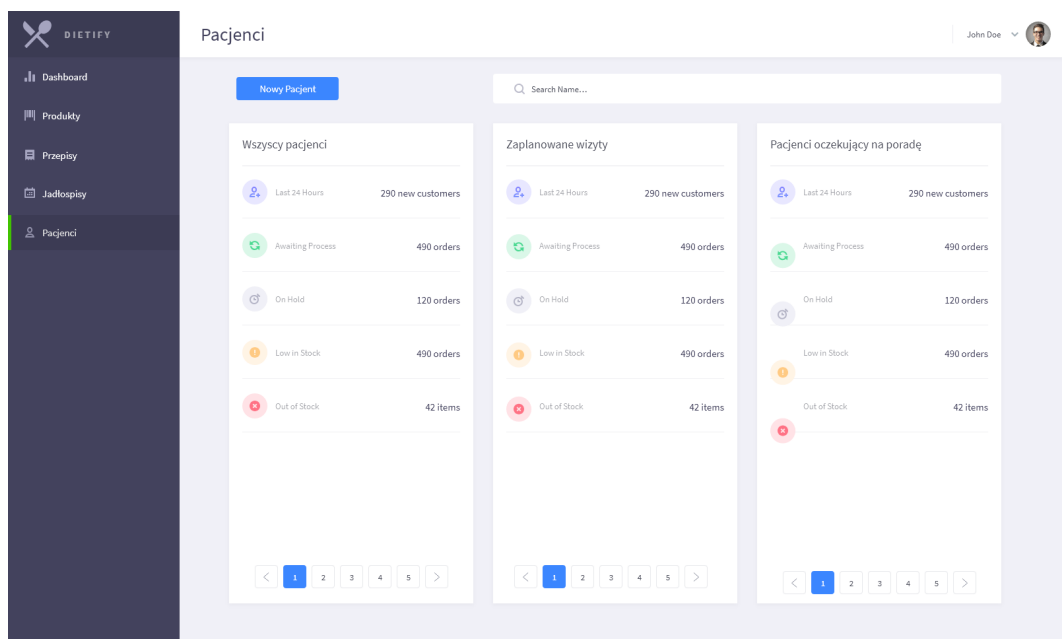


Rys. 4.15: Mockup11 (opr.wł).

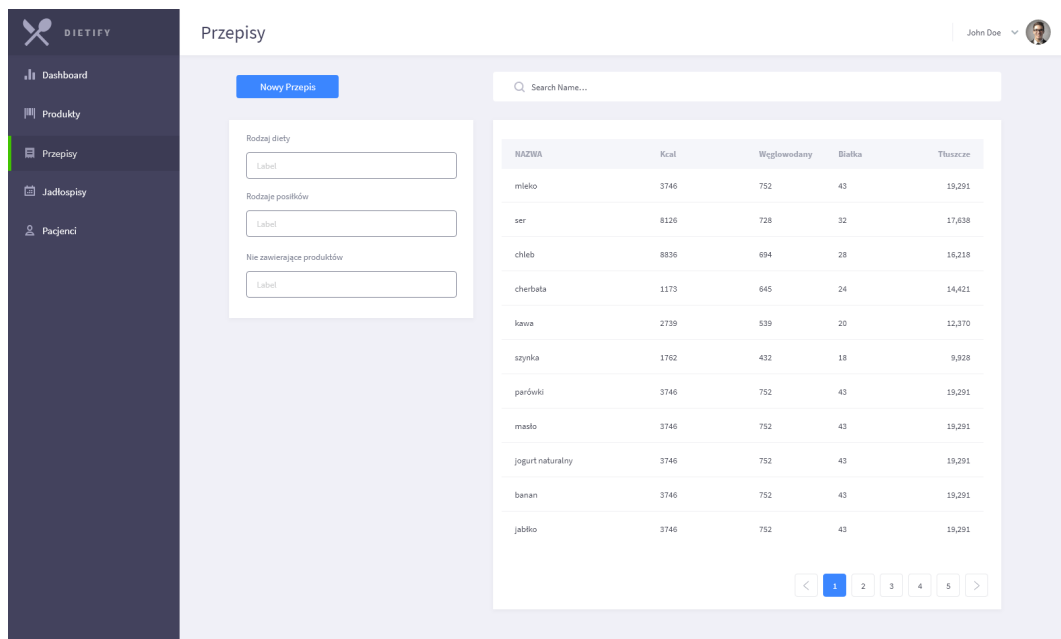




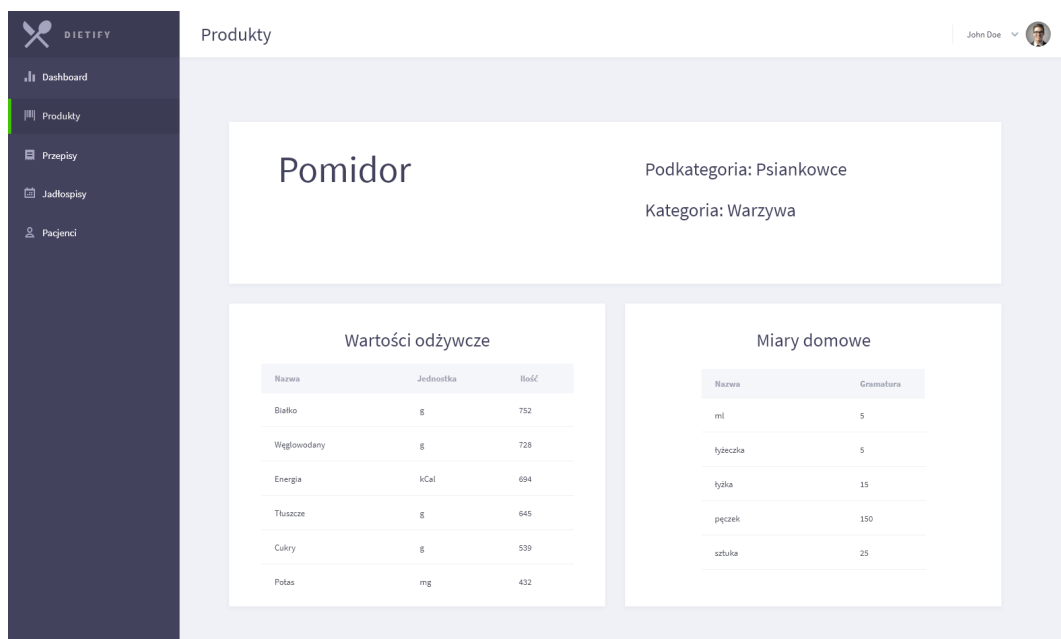
Rys. 4.16: Mockup12 (opr.wł).



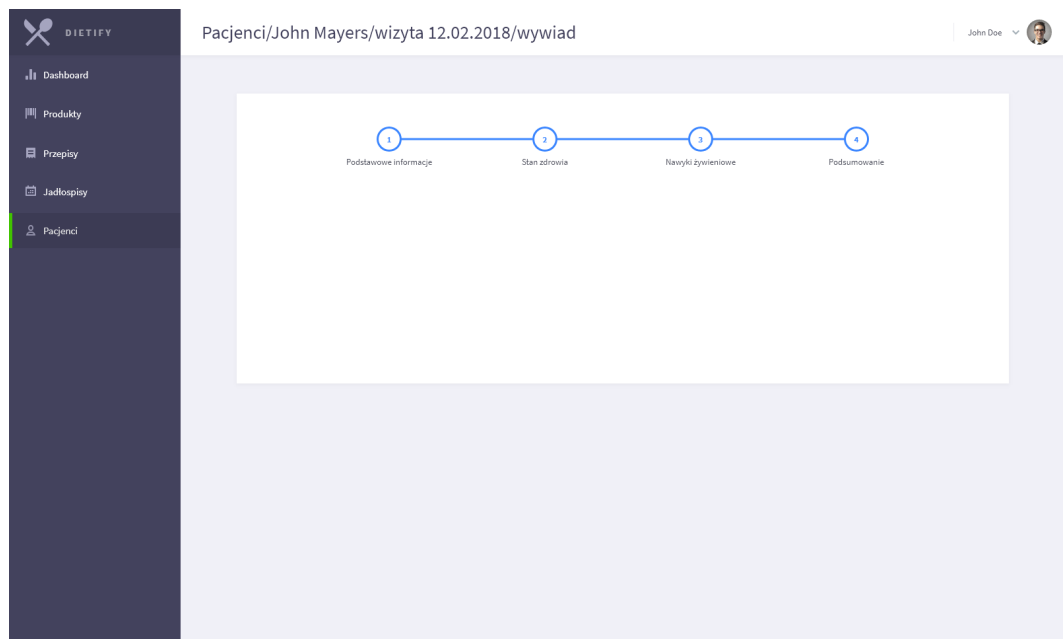
Rys. 4.17: Mockup13 (opr.wł).



Rys. 4.18: Mockup14 (opr.wł).



Rys. 4.19: Mockup15 (opr.wł).



Rys. 4.20: Mockup16 (opr.wł).

## 4.7. Baza danych

24.diagram erd 25.jdl 26.mikroserwisy

## 5. Implementacja

### 5.1. Wykorzystywane środowiska i narzędzia programistyczne

**27.uzupełnić, opisać, zacytować narzędzia** Podstawowe kryteria wyboru języka są następujące:

- Ścisła kontrola typów
  - Dobre wsparcie dla paradygmatu programowania obiektowego
  - Niezależność języka od platformy
- Wybrane przeze mnie języki spełniające te kryteria to:
- W warstwie backendu Java [5]
  - W warstwie frontendu Typescript [3]

#### 1. Backend

- Java [5]
- Spring [7]
- Maven [9]

#### 2. Frontend

- Typescript [3]
- Angular [2]

#### 3. Baza danych

- MySQL [6]
- Hibernate [8]
- Liquibase [1]

#### 4. Testy

- JUnit [10] /Mockito
- Protractor/Selenium

#### 5. Inne

- JHipster
- Gitlab

### 5.2. Architektura systemu

**28.opisać stack netflix oss 29.diagram rozmieszczenia**

### 5.3. Instalacja oprogramowania

**30.opisać instalację oprogramowania i warunki wstępne**

#### 5.3.1. Wymagania wstępne

- Java 8 [5]
- Node.js 10 [4]
- Docker + Docker Compose

### 5.4. Instrukcja użytkownika

**31.podstawowy opis poruszania się po aplikacji**

## **6. Testy**

### **6.1. Testy jednostkowe**

32.testy jednostkowe

### **6.2. Testy integracyjne**

33.testy integracyjne

### **6.3. Testy akceptacyjne**

34.testy akceptacyjne

## Zakończenie

### 6.4. Podsumowanie pracy

35.W pracy udało mi się dużo zrobić. Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

### 6.5. Możliwości dalszego rozwoju

36.Mnóstwo innych rzeczy da się poprawić i rozwinąć. Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

## Bibliografia

- [1] Datical, *Liquibase 3.6.3*, <https://www.liquibase.org/>. Dostęp 10.05.2019.
- [2] Google, *Angular 7.2.4*, <https://angular.io>. Dostęp 10.05.2019.
- [3] Microsoft, *Typescript 3.4*, <https://www.typescriptlang.org/>. Dostęp 10.05.2019.
- [4] Node.js Foundation, *Node.js 10.15.3 lts*, <https://nodejs.org/dist/latest-v10.x/docs/api/>. Dostęp 10.05.2019.
- [5] Oracle Corporation, *Java 8 se*, <https://docs.oracle.com/javase/8/docs/>. Dostęp 10.05.2019.
- [6] Oracle Corporation, *Mysql 8.0*, <https://www.mysql.com/>. Dostęp 10.05.2019.
- [7] Pivotal Software, *Spring framework 5*, <https://spring.io/>. Dostęp 10.05.2019.
- [8] Red Hat, *Hibernate 5.4*, <https://hibernate.org/>. Dostęp 10.05.2019.
- [9] The Apache Software Foundation, *Apache maven 3.6.1*, <https://maven.apache.org/>. Dostęp 10.05.2019.
- [10] The JUnit Team, *Junit 5*, <https://junit.org/junit5/>. Dostęp 10.05.2019.

## Spis rysunków

4.1. Produkty (opr.wł).	11
4.2. Przepisy (opr.wł).	11
4.3. Jadłospisy (opr.wł).	12
4.4. Wizyty (opr.wł).	12
4.5. Mockup1 (opr.wł).	13
4.6. Mockup2 (opr.wł).	13
4.7. Mockup3 (opr.wł).	14
4.8. Mockup4 (opr.wł).	15
4.9. Mockup5 (opr.wł).	15
4.10. Mockup6 (opr.wł).	16
4.11. Mockup7 (opr.wł).	16
4.12. Mockup8 (opr.wł).	17
4.13. Mockup9 (opr.wł).	17
4.14. Mockup10 (opr.wł).	18
4.15. Mockup11 (opr.wł).	18
4.16. Mockup12 (opr.wł).	19
4.17. Mockup13 (opr.wł).	19
4.18. Mockup14 (opr.wł).	20
4.19. Mockup15 (opr.wł).	20
4.20. Mockup16 (opr.wł).	21

# Spis tabel

1.1. Użytkownicy (opr.wł) . . . . .	3
3.1. Wymagania funkcjonalne ogólne (opr.wł) . . . . .	6
3.2. Wymagania funkcjonalne dla produktów (opr.wł) . . . . .	6
3.3. Wymagania funkcjonalne dla przepisów (opr.wł) . . . . .	7
3.4. Wymagania funkcjonalne dla jadłospisów (opr.wł) . . . . .	7
3.5. Wymagania funkcjonalne dla wizyt (opr.wł) . . . . .	8



## **Spis kodów źródłowych**

## A. To powinien być dodatek

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.