

<div>Projekt</div> <div>Inteligentne Usługi Informacyjne</div> <div>Wydział Elektrotechniki, Automatyki i Informatyki</div> <div>Politechnika Świętokrzyska</div>	
Studia: Stacjonarne II Stopnia	Kierunek: Informatyka
Grupa: 1ID22B	Skład zespołu: Zakrzewski Mateusz Zachariasz Łukasz
<div>Temat</div> <div>Opracowanie projektu i aplikacji systemu</div> <div>do sugerowania oraz zamawiania posiłków.</div>	

1. WSTĘP

Głównym celem projektu było wykonanie oprogramowania, które umożliwia dokonywanie zamówień spersonalizowanych dla konkretnego użytkownika ofert pojedynczych posiłków oraz zestawów.

Mechanizm sugerowania produktów bazuje na danych dotyczących pory dnia, dziennego zapotrzebowania kalorycznego użytkownika czy wskaźnika masy ciała – BMI.

System został zrealizowany w architekturze webowej aplikacji klient-serwer, wykorzystano *Framework Spring* oraz *Angular 8*. Na system składają się trzy odrębne aplikacje: **aplikacja Serwera** oraz **dwie aplikacje Klientkie**, z których jedna jest systemem do zarządzania zasobami z poziomu administratora, a druga końcowym serwisem dla użytkownika.

Wydzielone moduły systemu:

<i>Aplikacja Serwera:</i>	<i>food-order-back</i>
<i>Aplikacja dla Administratora:</i>	<i>food-order-front</i>
<i>Aplikacja dla Klienta:</i>	<i>food-order-store-front</i>

Rozdział drugi zawiera przedstawienie technologii wykorzystanych do stworzenia najważniejszych funkcjonalności projektu z wyodrębnieniem na część *back-end* oraz *front-end*.

Rozdział trzeci to opis architektury projektu oraz relacji pomiędzy komponentami systemu.

Kolejny, czwarty rozdział zawiera instrukcję obsługi i uruchomienia systemu.

W rozdziale piątym przedstawiona została struktura każdej z aplikacji składającej się na cały system, wyróżniono najważniejsze fragmenty struktury oraz przedstawiono ogólną organizację poszczególnych elementów.

Rozdział szósty przedstawia wszystkie funkcjonalności aplikacji od strony implementacyjnej.

W rozdziale siódmym zaprezentowano wszystkie istotne funkcjonalności systemu od strony użytkownika końcowego.

W ósmym rozdziale umieszczona została bibliografia, która zawiera najważniejsze materiały wykorzystane przy budowie projektu.

2. WYKORZYSTANE TECHNOLOGIE

2.1. Aplikacja Serwera

Aplikacja serwera w całości została napisana w języku obiektowym *Java*. Jako podstawę do budowy serwera wykorzystano *Spring Framework*, który znacznie przyspiesza i usprawnia budowanie oraz konfigurowanie aplikacji serwera.

Spring Framework dostarcza również wykorzystany *Spring Data*, który to ułatwia wykorzystanie mechanizmu *Java Persistence API* odpowiedzialnego za utrwalanie obiektów w bazie danych – do tego celu wykorzystywany jest *Hibernate*.

Projekt oparty został o narzędzie *Maven*, które pozwala dogodnie zarządzać zależnościami przypiętymi do projektu.

Dane przechowywane są w bazie *MySQL* w wersji 5.

Aplikacja serwera obsługuje również mechanizm sesji użytkowników. Do przechowywania sesji wykorzystano bazę typu *in-memory* *Redis*. Jest to szybko działająca rozproszona baza *klucz-wartość* przechowująca dane w pamięci operacyjnej.

Autoryzacja odbywa się na podstawie tokenu pobieranego z ID sesji HTTP.

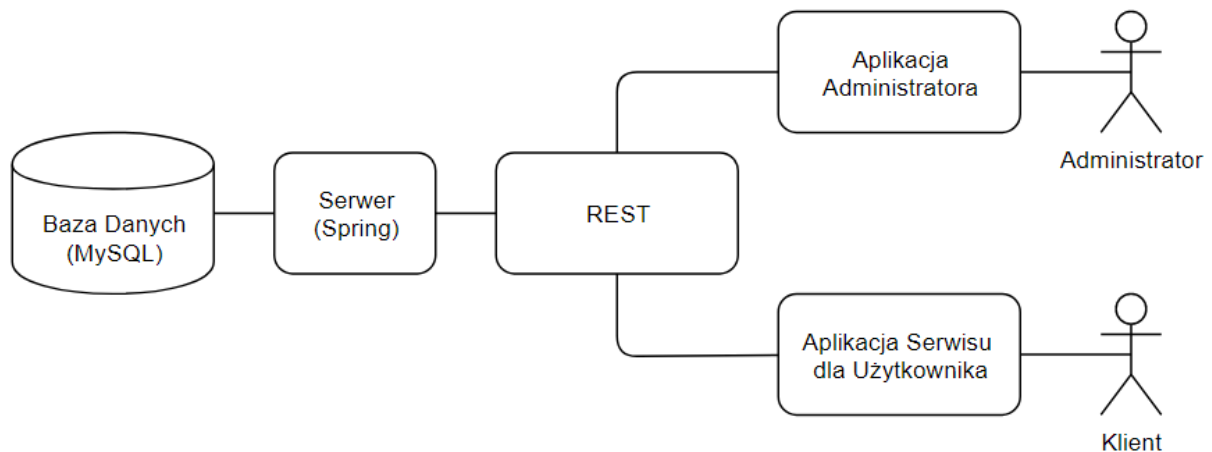
2.2. Aplikacje klienta

Aplikacje klienta zostały napisane w języku *TypeScript* z wykorzystaniem *frameworka Angular 8*.

Do komponentów UI wykorzystano *Angular Material* oraz *Bootstrap*. Biblioteki te zapewniły dostęp do najbardziej niezbędnych elementów, takich jak paski nawigacyjne, przyciski, pola tekstowe. Dzięki wykorzystaniu powyższych aplikacja jest stworzona w sposób responsywny.

3. ARCHITEKTURA APLIKACJI

Aplikacje klienta zbudowane zostały w architekturze REST – wszystkie funkcjonalności oparte są o odpowiedzi z protokołu HTTP. Wykorzystano wzorzec MVC (*Model – View – Controller*). Schemat relacji poszczególnych elementów składających się na system prezentuje poniższy diagram.

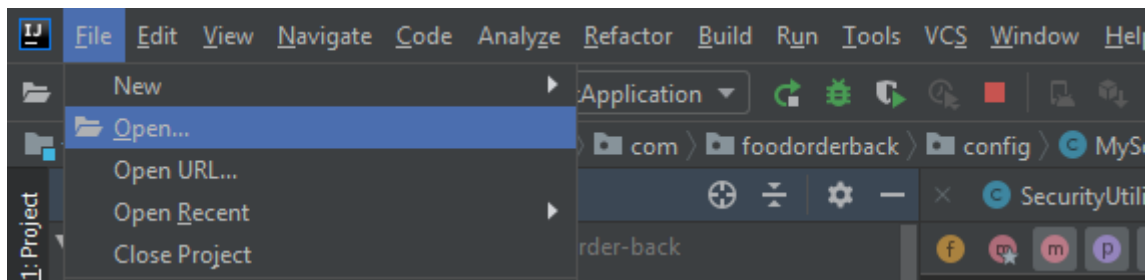


Zarówno Klient jak i Administrator wykorzystują wystawione przez Serwer REST API. Aplikacja Serwera przetwarza odebrane informacje wysyłając odpowiednie odpowiedzi poprzez protokół HTTP. Odebrane odpowiedzi subskrybowane są przez odpowiednie *service'y* aplikacji *frontend'owych* a następnie odbierane przez komponenty przeznaczone do prezentacji i obsługi odebranych danych.

4. INSTRUKCJA URUCHOMIENIA ORAZ KONFIGURACJA

4.1. Aplikacja serwera

Do sprawozdania dołączono archiwum ZIP z projektem a także link do repozytorium zdalnego GitHub. Uruchomienie projektów poszczególnych aplikacji można wykonać na dwa sposoby. Pierwszym jest rozpakowanie paczki ZIP z projektami a następnie wskazanie poszczególnych z nich w środowisku IDE InteliJ IDEA, drugim sklonowanie repozytorium. Aby otworzyć projekt należy wybrać opcję *File -> Open* i wskazać katalog projektu jak pokazano na obrazku poniżej.



Po otwarciu projektu środowisko zaimportuje konfigurację projektu *Maven* i automatycznie pobierze zależności wymagane do działania aplikacji. Aplikacja wymaga JDK w wersji 1.8.

Przed uruchomieniem aplikacji należy zainstalować bazę **MySQL w wersji 5** oraz pobrać bazę Redis'a w wersji: **Redis 2.8.2402 64 bit**.

Po instalacji MySQL należy dodać bazę o nazwie: ***iui_db***.

Aby uruchomić bazę Redis należy rozpakować archiwum ZIP dostępne pod adresem: <https://github.com/MicrosoftArchive/redis/releases/tag/win-2.8.2402>

a następnie w głównym katalogu rozpakowanego archiwum otworzyć okno terminala (konsoli) i wywołać następującą komendę: ***redis-server.exe --maxheap 1GB***

Po wykonaniu komendy powinien pojawić się zbiór komunikatów postaci przedstawionej na poniższym zrzucie ekranu.

```
C:\Windows\System32\cmd.exe - redis-server.exe --maxheap 1GB
C:\Users\tukasz\Desktop\REDIS>redis-server.exe --maxheap 1GB

Redis 2.8.2402 (00000000/0) 64 bit
Running in stand alone mode
Port: 6379
PID: 9472

http://redis.io

[9472] 29 Jan 15:52:04.420 # Server started, Redis version 2.8.2402
[9472] 29 Jan 15:52:04.423 * The server is now ready to accept connections on port 6379
```

Flaga ***--maxheap*** określa maksymalny rozmiar bazy tworzonej w pamięci RAM.

Po utworzeniu bazy *MySQL* oraz uruchomieniu *Redis'a* można uruchomić aplikację serwera klikając **zieloną strzałkę** w panelu *IntelliJ IDEA*. Na starcie serwer doda domyślnego użytkownika do bazy z rolą Administratora o następujących danych autoryzacyjnych:

Login: Lukasz **Hasło:** superhaslo

Pomyślne uruchomienie serwera prezentuje zestaw logów w oknie konsoli *IntelliJ IDEA* informujących, że aplikacja wystartowała, co prezentuje poniższy obrazek.

```
2020-01-29 18:59:14.848 INFO 13592 --- [main] org.hibernate.Version : HHH0000412: Hibernate Core (5.3.10.Final)
2020-01-29 18:59:14.849 INFO 13592 --- [main] org.hibernate.cfg.Environment : HHH0000206: hibernate.properties not found
2020-01-29 18:59:14.972 INFO 13592 --- [main] org.hibernate.annotations.common.Version : HCA000000001: Hibernate Commons Annotations {5.0.4.Final}
2020-01-29 18:59:15.113 INFO 13592 --- [main] org.hibernate.dialect.Dialect : HHH0000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
2020-01-29 18:59:17.265 INFO 13592 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2020-01-29 18:59:18.064 INFO 13592 --- [main] io.lettuce.core.EpollProvider : Starting without optional epoll library
2020-01-29 18:59:18.065 INFO 13592 --- [main] io.lettuce.core.KqueueProvider : Starting without optional kqueue library
2020-01-29 18:59:18.549 INFO 13592 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.security.web.context.
2020-01-29 18:59:18.748 INFO 13592 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-01-29 18:59:18.798 WARN 13592 --- [main] s.a.ScheduledAnnotationBeanPostProcessor : spring.jpa.open-in-view is enabled by default. Therefore, database queries may
2020-01-29 18:59:19.725 INFO 13592 --- [main] s.a.ScheduledAnnotationBeanPostProcessor : No TaskScheduler/ScheduledExecutorService bean found for scheduled processing
2020-01-29 18:59:19.749 INFO 13592 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-01-29 18:59:19.751 INFO 13592 --- [main] c.f.FoodOrderBackApplication : Started FoodOrderBackApplication in 8.84 seconds (JVM running for 10.002)
2020-01-29 18:59:20.296 INFO 13592 --- [main] o.h.h.i.QueryTranslatorFactoryInitiator : HHH0000397: Using ASTQueryTranslatorFactory
2020-01-29 18:59:20.375 INFO 13592 --- [main] c.f.s.i.UserManagementService : Already in database Lukasz
```

4.2. Aplikacje klienta

Z uwagi na fakt, że obie aplikacje klienckie stworzone są w podobnej architekturze, z wykorzystaniem Angular CLI sposób uruchomienia ich będzie przebiegał w ten sam sposób, zostanie więc przedstawiony sposób uruchomienia na przykładzie aplikacji serwisu dla użytkownika końcowego.

Zanim nastąpi przejście do uruchomienia należy zainstalować niezbędne środowisko *NodeJS*, które można uzyskać pod adresem: <https://nodejs.org/>

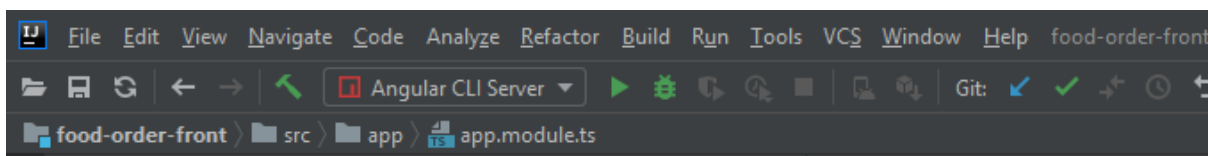
W analogiczny sposób należy wskazać w kolejnej otwartej instancji środowiska *IntelliJ* projekt aplikacji *food-order-store-front* lub *food-store-front* (w zależności, którą aplikację chcemy uruchomić z klienckich).

Po zaindeksowaniu wybranego projektu oraz pobraniu automatycznym bibliotek można przejść do uruchomienia jednak zaleca się wykonanie komendy *npm install* w przypadku gdy zależności nie zastały przez środowisko automatycznie pobrane.

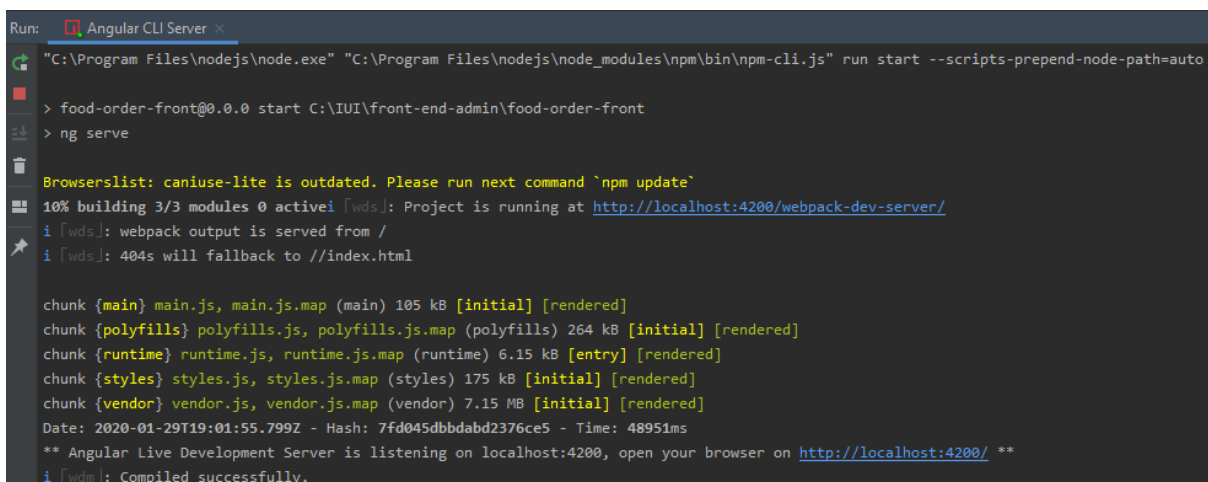
Aby uruchomić aplikację należy wykonać w panelu terminala IDE komendę: *ng serve*

natomiast jeśli chcemy uruchomić aplikację na odrębnym porcie lub dostępną z adresu IP w lokalnej sieci można wykorzystać składnię: *ng serve --host adres ip --port nr portu*

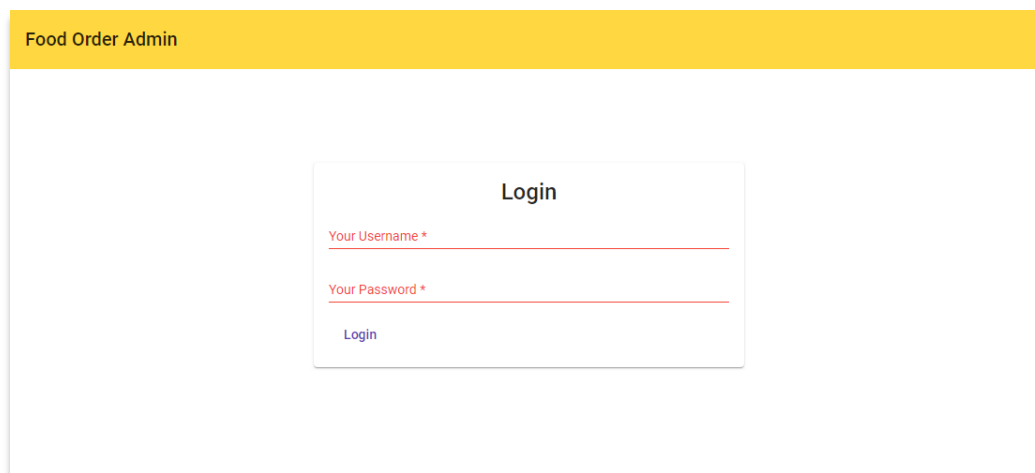
Innym sposobem uruchomienia aplikacji jest po rozpoznaniu przez IDE projektu Angular CLI wciśnięcie zielonej strzałki startującej projekt.



Po uruchomieniu projektu w oknie terminala ujrzymy informację o pomyślnym wystartowaniu projektu oraz pod jakim adresem jest dostępna, w tym przypadku będzie to host lokalny, jak przedstawia poniższy zrzut ekranu.



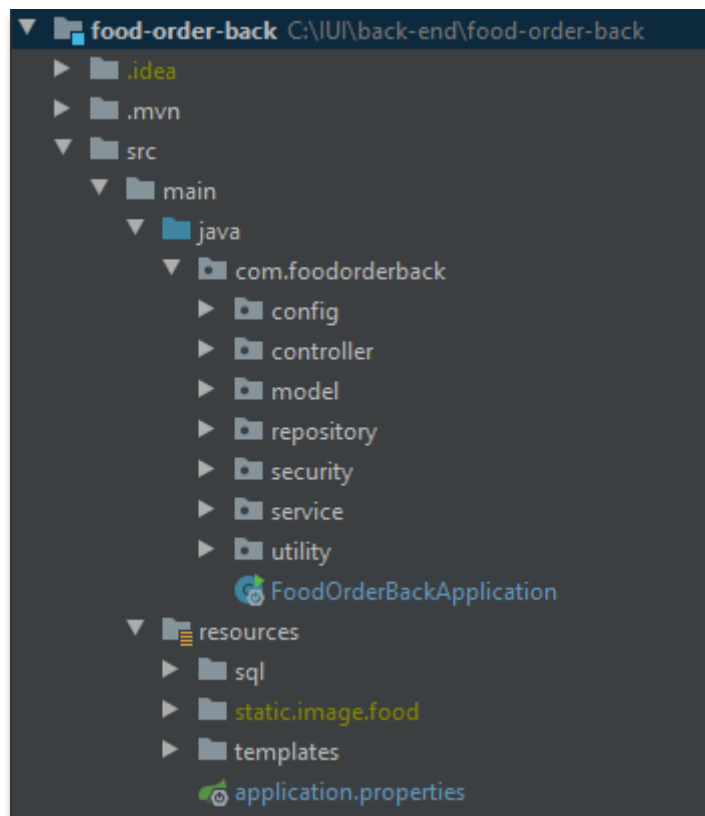
Pod podanym URL załaduje się główna strona projektu aplikacji panelu administratora.



5. STRUKTURA PROJEKTÓW APLIKACJI

5.1. Aplikacja serwera (*food-order-back*)

Strukturę wewnętrznej organizacji projektu przedstawiono na poniższym obrazku, który prezentuje podział na pakiety odpowiedzialne za poszczególne zasoby i implementacje funkcji serwera.

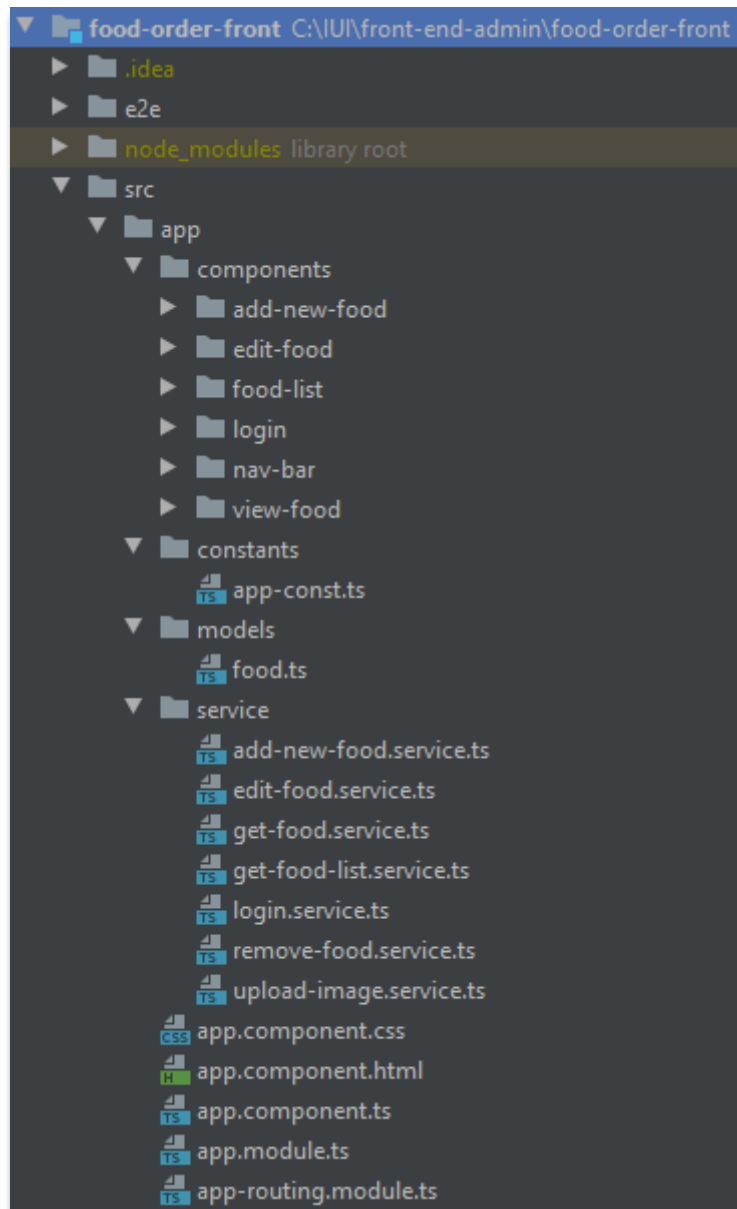


- W pakiecie **config** znajdują się klasy odpowiedzialne za konfigurację aplikacji serwera. Konfiguracja dotyczy obsługi sesji użytkowników z pomocą *Redis'a*, autoryzacji sesji dla podanych URL, filtrów *Requestów* oraz generowania hasła dla nowych utworzonych kont.
- W pakiecie **controller** znajdują się klasy udostępniające wszystkie punkty końcowe odpowiedzialne za wszystkie funkcjonalności dostępne przez RESTApi.
- W pakiecie **model** zawierają się wszystkie obiekty reprezentujące podstawowe obiekty na których wykonywane są operacje. Są to wzorce dla bazy danych, na podstawie których tworzone są w niej encje. Obiekty modelu zawierają opisane relacje między sobą za pomocą odpowiednich adnotacji.

- W pakiecie **repository** zdefiniowane zostały wszystkie repozytoria dla obiektów przechowywanych w bazie danych. Są to *interfejsy* implementujące interfejs CRUDRepository dla odpowiedniego dla siebie typu danych.
- W pakiecie **security** umieszczone zostały klasy obiektów związanych z autoryzacją i przyznawaniem dostępu do zasobów serwisu i *end-point'ów* REST'owych. W pakiecie tym umieszczono również obiekt opisujący Role użytkowników (ROLE_ADMIN oraz ROLE_USER) oraz obiekt UserRole łączący rolę z użytkownikiem.
- W pakiecie **service** analogicznie do nazwy umieszczone zostały klasy dokonujące operacji na repozytoriach oraz wykonujące i zawierające istotne implementacje funkcjonalności w tym sugerowanie posiłków, składanie zamówień czy obsługę autoryzacji. Każdy service został opatrzony definicją metod w dedykowanym dla niego interfejsie.
- W pakiecie **utility** umieszczono definicję obiektu, który jest odpowiedzialny za wysyłanie wiadomości email do użytkowników o rejestracji, przypomnieniu hasła, oraz podsumowaniu zamówienia po jego złożeniu.
- W folderze **resources** znajdują się pliki graficzne dodanych przez administratora posiłków oraz szablony dla wysyłanych wiadomości email, które zostały stworzone z wykorzystaniem *Thymeleaf'a*.

5.2. Aplikacja administratora serwisu (*food-order-front*)

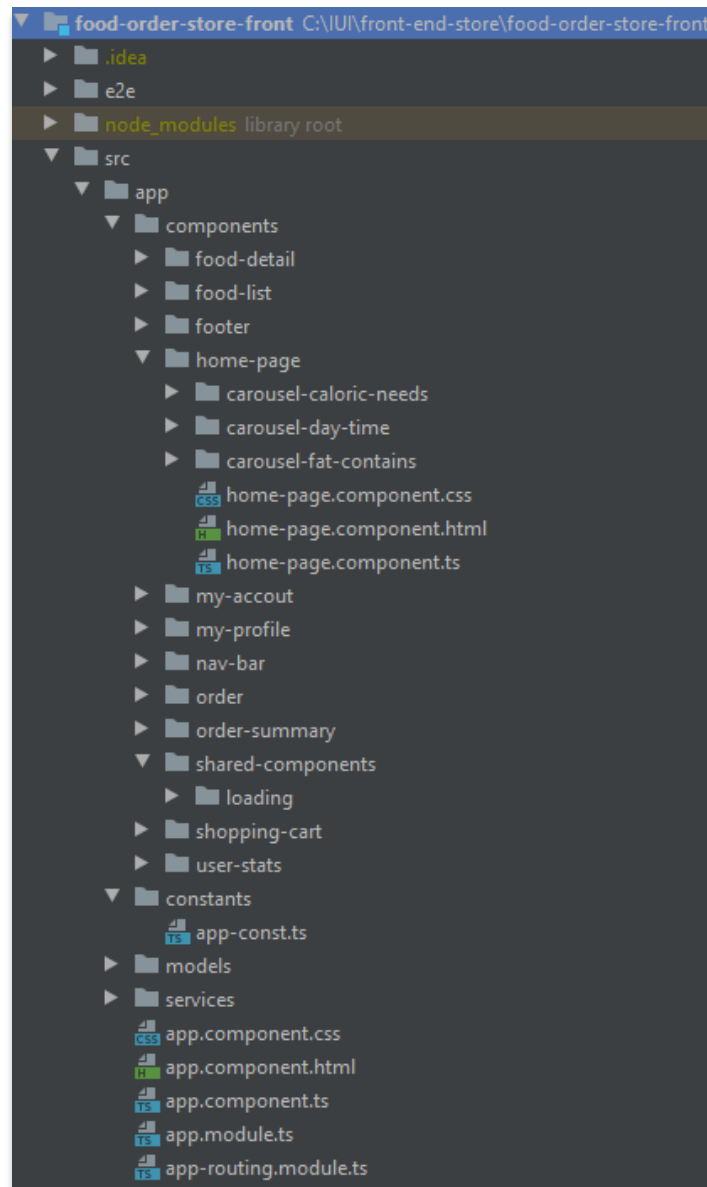
Struktura projektu aplikacji panelu administratora została podzielona ze względu na komponenty *angular-owe*, stałe wykorzystane w aplikacji, obiekty modeli danych, oraz *service'y*. Organizację plików w aplikacji prezentuje umieszczona poniżej struktura.



Jak prezentuje powyższa struktura dla aplikacji administratora można wyróżnić widoki dodawania do systemu nowego posiłku, jego edycję, wyświetlenie listy wszystkich dostępnych w systemie produktów. Dodatkowo przedstawiono w pakiecie **service** wszystkie *service'y* odpowiedzialne za obsługę i korzystanie z udostępnionego RESTApi.

5.3. Aplikacja serwisu dla użytkownika (*food-order-store-front*)

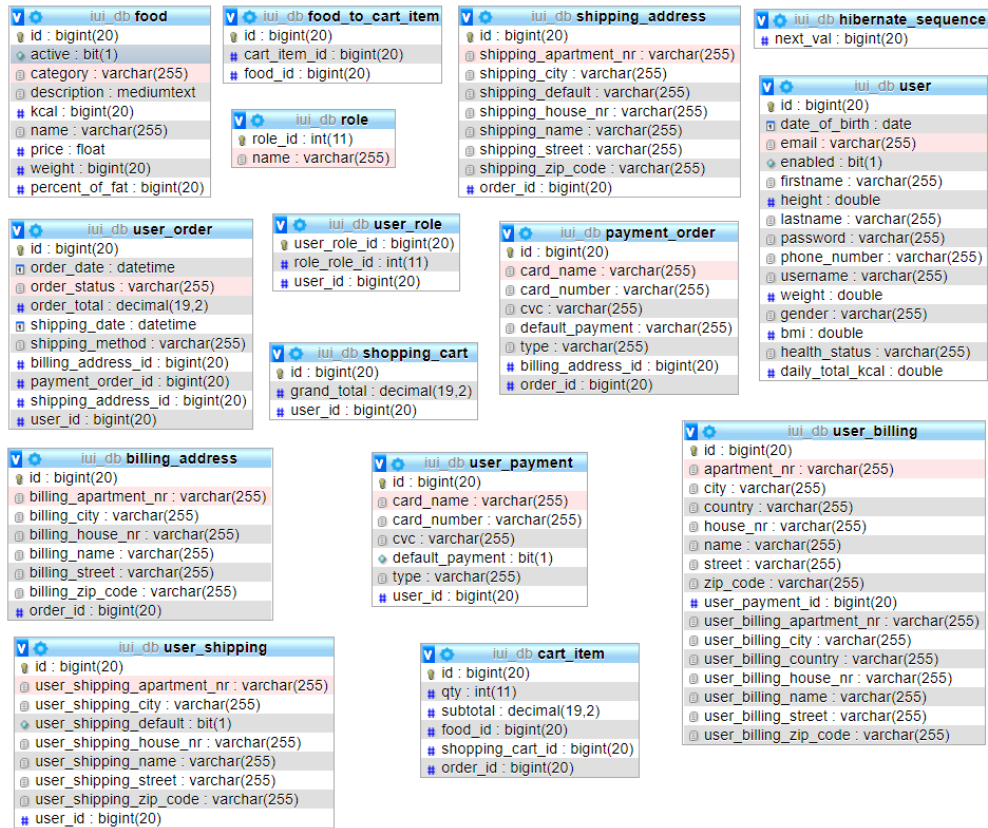
Podobnie w przypadku aplikacji serwisu dla użytkownika struktura zachowuje te same zasady organizacji, które przedstawione zostały odnośnie aplikacji administratora. Wydzielono osobno komponenty, stałe, *service*'y i modele.



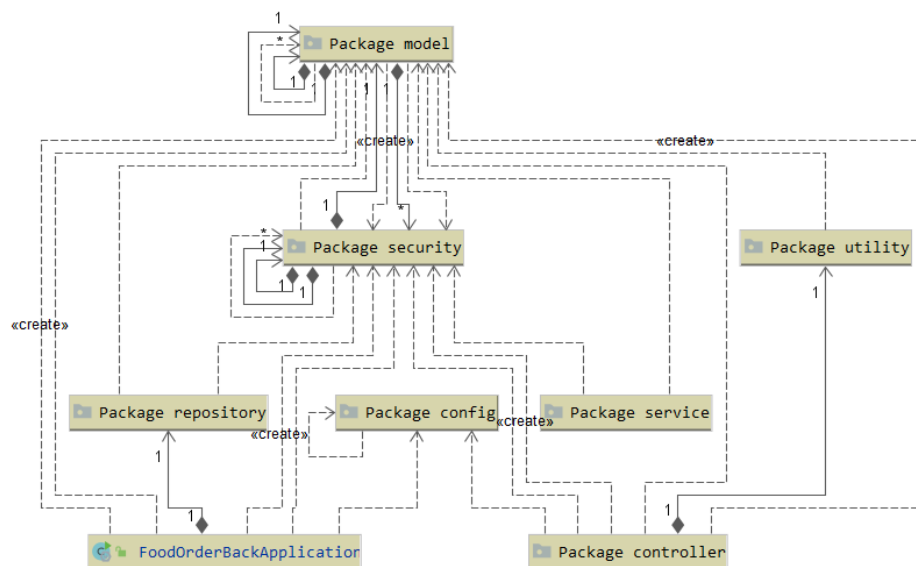
Komponenty zostały podzielone na szereg poszczególnych odpowiedzialnych za odrębne funkcjonalności takie jak m.in.: zarządzanie profilem użytkownika, zakładanie nowego konta, obsługę karty zamówienia, komponent strony domowej obsługuje wyświetlanie sugerowanych posiłków, komponent ten posiada zagnieżdżone komponenty z których każdy z nich kolejno sugeruje posiłki zgodnie z kryteriami przedstawionymi w kolejnym rozdziale.

6. SCHEMATY DO PROJEKTU

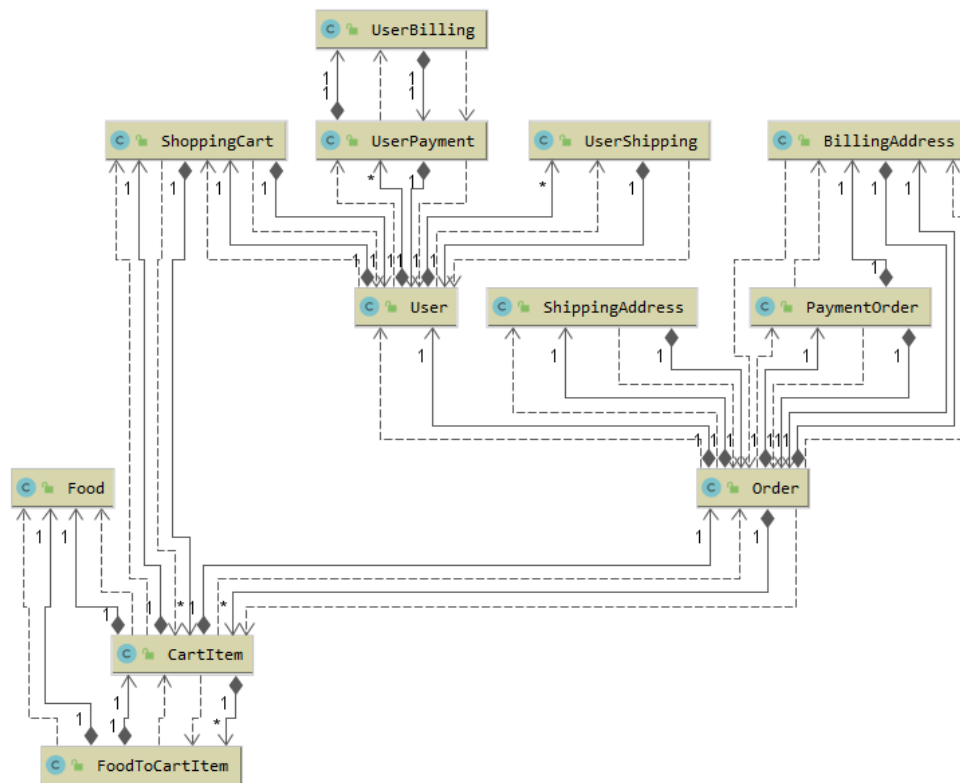
6.1. Stworzone encje



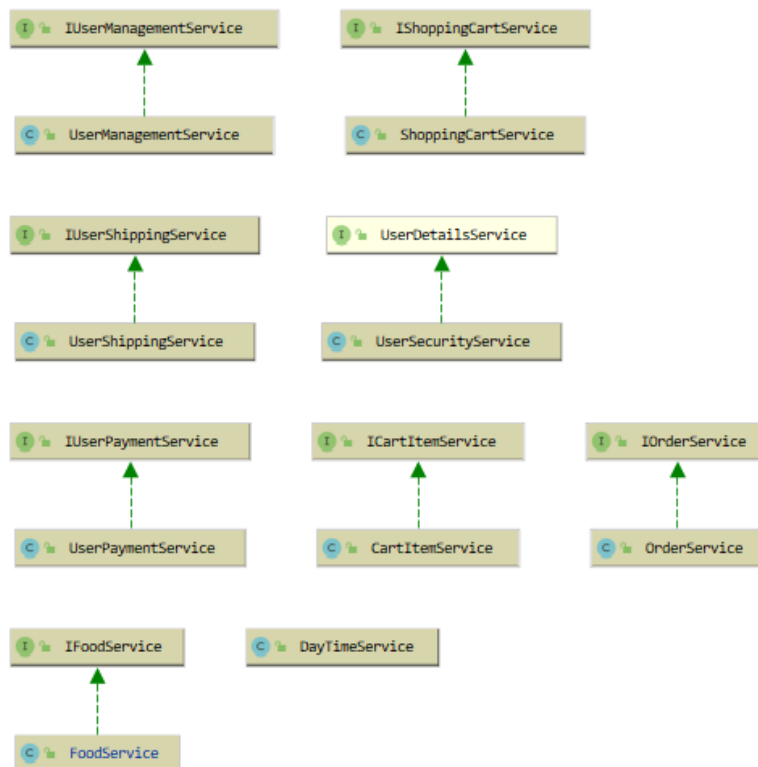
6.2. Zależności między pakietami



6.3. Powiązania między obiektami pakietu modelu



6.4. Dostępne Service'y



7. ZAIMPLEMENTOWANE FUNKCJONALNOŚCI

7.1. Spis najważniejszych funkcjonalności

Głównym celem było zaimplementowanie sugerowania posiłków dla zalogowanego użytkownika przez system, który na podstawie uzupełnionych danych takich jak: **wzrost, waga** czy **wiek** wyznaczał odpowiednie wskaźniki takie jak na przykład **BMI**. Odnosząc się do tych wyników system według zadanych kryteriów dobiera zestawy posiłków ze względu na:

- Dzielne zapotrzebowanie energetyczne (*carousel-caloric-needs*)
- Porę dnia (*carousel-day-time*)
- Wskaźnik masy ciała - BMI (*carousel-fat-contain*)

System wyznacza czy użytkownik ma prawidłową masę ciała, nadwagę lub niedowagę, na tej podstawie dobiera mniej lub bardziej kaloryczne posiłki, lub te mniej lub bardziej bogate w zawartość tłuszczu. Z uwagi na przeznaczenie serwisu zostały zaimplementowane również następujące funkcjonalności:

Dotyczące konta użytkownika:

- Zakładanie nowego konta
- Logowanie / Autoryzacja (Zarządzanie sesją użytkowników)
- Funkcja przypomnienia hasła
- Zarządzanie danymi konta
- Zarządzanie metodami płatności
- Zarządzanie adresami dostawy

Zarządzanie produktami:

- Dodawanie / Usuwanie / Edycja / Wyświetlanie posiłku.
- Logowanie / Wylogowanie / Autoryzacja / Zarządzanie sesją użytkowników
- Funkcja przypomnienia hasła

Dokonywanie zamówień:

- Dodawanie / Usuwanie / Modyfikacja koszyka produktów
- Zgłoszenie zamówienia
- Wysyłanie wiadomości email z potwierdzeniem i podsumowaniem

7.2. Implementacja najważniejszych funkcjonalności

Poniżej znajduje się metoda wyznaczająca najbardziej odpowiedni produkt dla zadanej pory dnia według kategorii posiłku oraz jak najbliższej wartości kalorycznej przeznaczonej na ten posiłek dla zalogowanego użytkownika uwzględniając jego całkowite zapotrzebowanie kaloryczne w ciągu dnia.

```
1. private Food findBestDishForDayTime(String dish,
2.                                     double bestCaloriesAmount) {
3.     Food bestDish = new Food();
4.     ArrayList<Food> allFoodCategory = (ArrayList<Food>)
5.     foodRepository.findAllByCategory(dish);
6.     if (allFoodCategory.size() > 0) {
7.         bestDish = allFoodCategory.get(0);
8.         double min = (double) bestDish.getKcal() * (double)
9.         (bestDish.getWeight() / 100d);
10.        double diff = Math.abs(min - bestCaloriesAmount);
11.        for (Food food : allFoodCategory) {
12.            double totalFoodEnergy = (double)
13.            food.getKcal()
14.            * (double) (food.getWeight() / 100d);
15.            if (Math.abs(totalFoodEnergy
16.            - bestCaloriesAmount)
17.            < diff) {
18.                bestDish = food;
19.            }
20.        }
21.        return bestDish;
22.    }
23.    return new Food();
24. }
```

Metoda zwracająca posiłki ze względu na kategorię według obecnej pory dnia

```
1. @Override
2.     public List<Food> getFoodByDayTime() {
3.         if (dayTimeService.isMorning()) {
4.             return foodRepository.findAllByCategory(BREAKFAST); }
5.         if (dayTimeService.isNoon()) {
6.             return foodRepository.findAllByCategory(DINNER); }
7.         if (dayTimeService.isAfterNoon()) {
8.             return foodRepository.findAllByCategory(LUNCH); }
9.         if (dayTimeService.isEvening()) {
10.            return foodRepository.findAllByCategory(SUPPER); }
11.        return new ArrayList<>();
12.    }
```

Pory dnia według założeń systemu trwają odpowiednio:

- Do godziny 10:00 - **Śniadanie**
- Od godziny 10:00 Do godziny 14:00 – **Obiad**
- Od godziny 14:00 Do godziny 17:00 – **Podwieczorek**
- Od godziny 17:00 – **Kolacja**

Kaloryczność każdego z posiłków w ciągu dnia w wymiarze procentowym względem całkowitego zapotrzebowania kalorycznego:

- **Śniadanie** – 30% zapotrzebowania
- **Obiad** – 35% zapotrzebowania
- **Podwieczorek** – 25% zapotrzebowania
- **Kolacja** – 10 % zapotrzebowania

Całkowite zapotrzebowanie kaloryczne liczone jest na podstawie wzoru metodą Mifflin-St Jeor. Wzór oraz implementacja przedstawiona została poniżej.

Mężczyźni

$$[9,99 \times \text{masa ciała (kg)}] + [6,25 \times \text{wzrost (cm)}] - [4,92 \times \text{wiek (lata)}] + 5$$

Kobiety

$$[9,99 \times \text{masa ciała (kg)}] + [6,25 \times \text{wzrost (cm)}] - [4,92 \times \text{wiek(lata)}] - 161$$

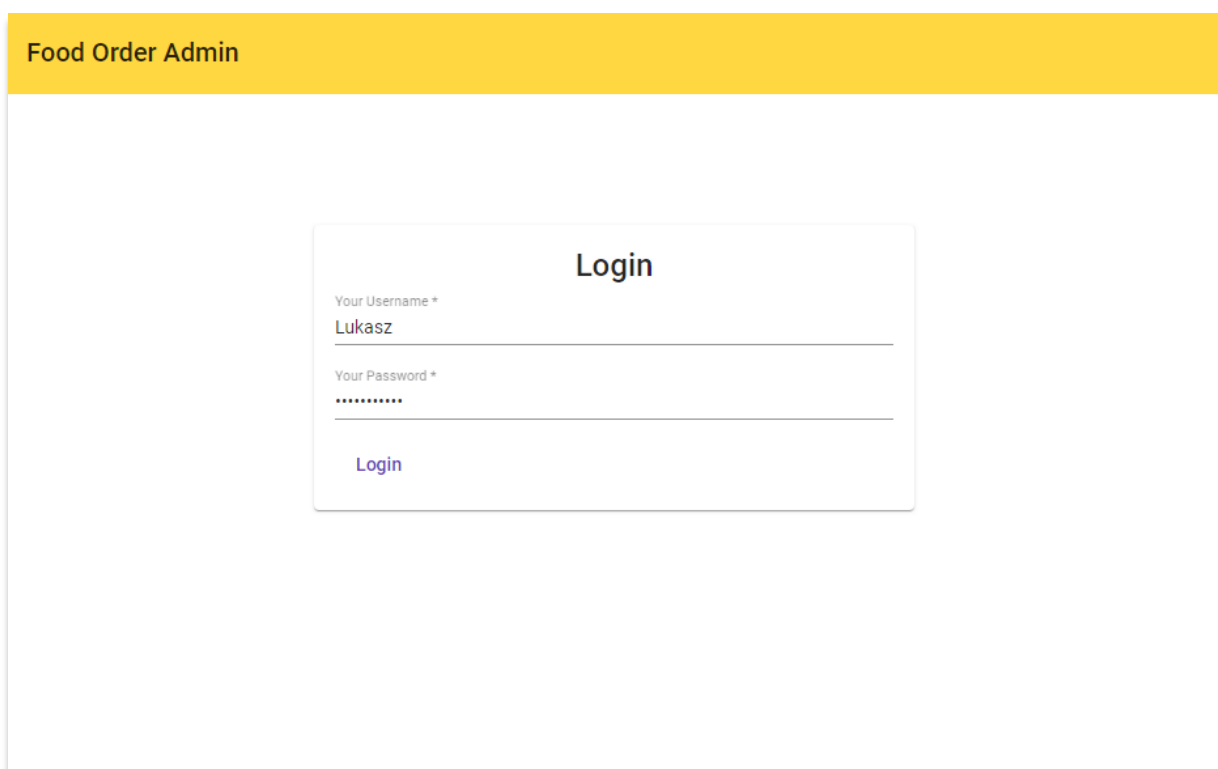
```
1. int age = Period.between(currentUser.getDateOfBirth()  
2. .toInstant()  
3. .atZone(ZoneOffset.UTC).toLocalDate(),  
4.     LocalDate.now()).getYears();  
5.  
6.     double v = 9.9d * weight + 6.25d * height - 4.92d * age;  
7.     if (gender.equals("Male"))  
8.         currentUser.setDailyTotalKcal(v + 5d);  
9.     if (gender.equals("Female"))  
10.         currentUser.setDailyTotalKcal(v - 161d);
```


8. PREZENTACJA WIZUALNA I FUNKCJONALNA APLIKACJI

W tym rozdziale przedstawiono wygląd oraz rozmieszczenie poszczególnych funkcji aplikacji administratora oraz klienta jako użytkownika końcowego. Przedstawiono efekty działań najważniejszych z funkcjonalności.

8.1. Aplikacja Panelu Administratora

Uruchamiając aplikację administratora i wchodząc pod adres strony głównej otrzymujemy widok z oknem logowania.



Food Order Admin

Login

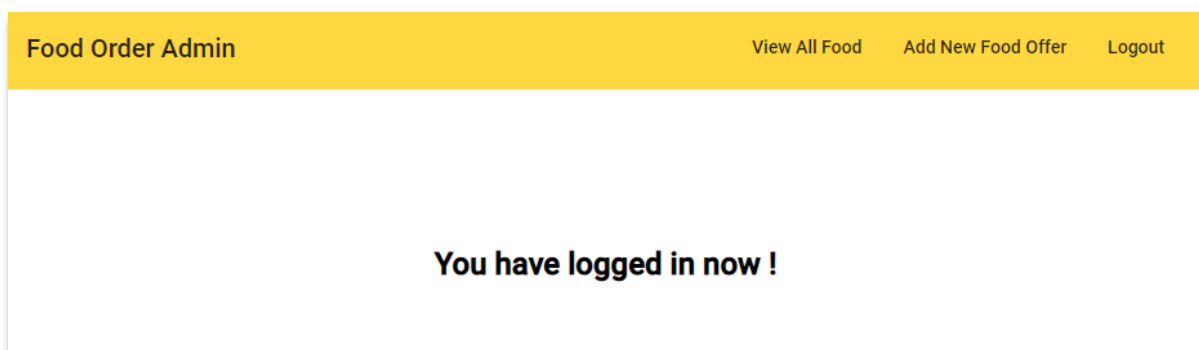
Your Username *

Lukasz

Your Password *

Login

Po zalogowaniu się otrzymujemy dostęp do funkcji wyświetlania listy z produktami oraz możliwością dodania nowego produktu.



Food Order Admin

View All Food Add New Food Offer Logout

You have logged in now !

<input checked="" type="checkbox"/>	Id	Name	Weight (g)	Energy (kcal/100g)	Fat (%)	Category	Price(PLN)	Status	Active	Action
<input checked="" type="checkbox"/>	223	Zupa przepomidorowa	100	430	43	Dinner	12	true		Delete
<input checked="" type="checkbox"/>	224	Pierogi	100	170	99	Supper	32	true		Delete
<input checked="" type="checkbox"/>	225	Flaki	100	510	23	Breakfast	43	true		Delete
<input checked="" type="checkbox"/>	359	Jakieś jedzonko	2342	1234	100	Lunch	2344	true		Delete
<input checked="" type="checkbox"/>	412	Fit Kulki Z Suszonymi Jabłkami I Cynamonem	175	301	4	Supper	19.99	true		Delete

[Remove Selected](#)

Widok z listą przykładowo dodanych produktów, lista daje możliwość usuwania pojedynczego produktu lub wielu, po kliknięciu na dany produkt, strona przenosi administratora na kartę produktu, którą prezentuje kolejny zrzut ekranu.

Fit Kulki Z Suszonymi Jabłkami I Cynamonem

**Weight (g):** 175**Energy (kcal/100g):** 301**Category:** Supper**Price (PLN):** 19.99**Fat (%):** 4**Available:** true

Kulki z rodzynkami, daktylami, migdałami, orzechami laskowymi i kawałkami suszonych jabłek z cynamonem i płatkami owsianymi. Jedna porcja składa się z 5 kulek.

[Go Back](#)[Edit](#)

Z poziomu widoku kart produktu można przejść do edycji posiłku za pomocą przycisku **Edit**. Edycja umożliwia zmianę wszystkich podstawowych cech produktu oraz aktualnego zdjęcia. Widok edycji przedstawiono poniżej.

The screenshot shows the 'Edit your food food' form within the 'Food Order Admin' interface. The form contains the following fields and controls:

- Food Name ***: A text input field containing 'Fit Kulki Z Suszonymi Jabłkami I Cynamonem'.
- Describe this food ***: A text area containing 'Kulki z rodzynkami, daktylami, migdałami, orzechami laskowymi i kawałkami suszonych jabłek z cynamonem i płatkami owsianymi. Jedna porcja składa się z 5 kulek.' with a small edit icon (//) on the right.
- Energy (kcal/100g) ***: A text input field containing '301'.
- Weight (g) ***: A text input field containing '175'.
- Percentage Of Fat**: A slider control with a yellow marker at the beginning.
- Insert Food Price (PLN) ***: A text input field containing '19,99'.
- Select category ***: A dropdown menu with 'Supper' selected.
- Available to order**: A toggle switch that is currently turned on (yellow).
- Upload**: A button for uploading an image, located below the toggle switch.
- Update Food Order**: A button at the bottom left of the form.
- Cancel**: A red button at the bottom right of the form.

Analogicznym do Edycji jest panel dodawania produktu po przejściu do niego klikając pozycję z paska nawigacyjnego **Add New Food Offer**.

Administrator po zakończeniu pracy może wylogować się z panelu wybierając przycisk **Logout** umieszczony na pasku nawigacyjnym.

8.2. Aplikacja Serwisu Użytkownika Końcowego

Uruchamiając aplikację serwisu użytkownika końcowego i wchodząc pod adres strony głównej otrzymujemy widok z oknem zarządzania dostępem do serwisu.

Food Order Store My Account

Login, Register or get forgotten password

Login to your account

Username

Password

SIGN IN

Register new account

I forgot my password

UI - Zachariasz - Zakrzewski - 11022B - © 2020 Copyright

Użytkownik może zalogować się na istniejące konto, przypomnieć hasło otrzymując nowe na podany adres email lub zarejestrować nowe konto jeśli wcześniej tego nie zrobił. Podczas tworzenia konta należy podać adres email oraz login. Po podaniu tych danych do użytkownika zostanie wysłany email z wygenerowanym hasłem, które użytkownik może później zmienić na własne. Rejestrację oraz wysłany email prezentują poniższe zrzuty ekranu.

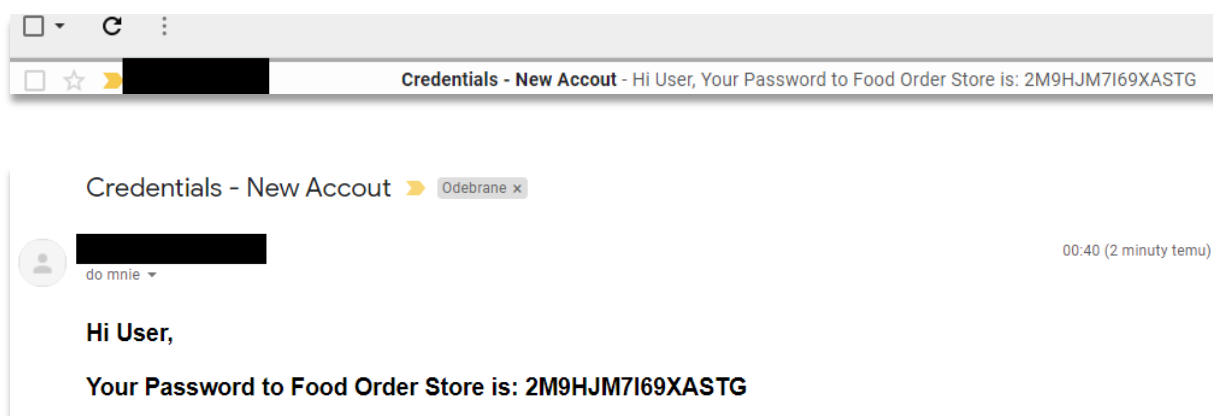
Register new account

User

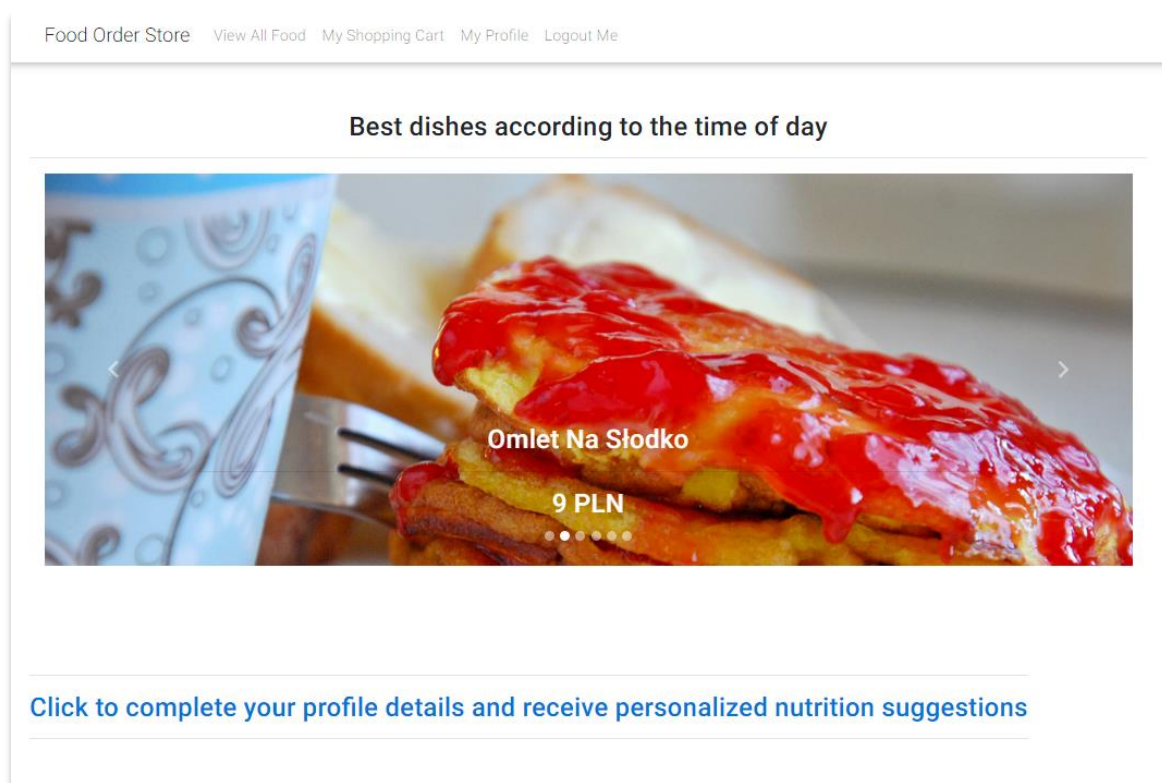
REGISTER

Register has been made. Email with password has been sent.

Zrzuty ekranu z *Gmaila*, na który otrzymano wygenerowane hasło.



W tym momencie po zalogowaniu się na nowo utworzone konto, nowy użytkownik otrzyma sugestie posiłków jedynie ze względu na porę dnia. Na stronie głównej wyświetlony jest komunikat, który mówi, że w celu otrzymania bardziej spersonalizowanych propozycji należy uzupełnić dane profilu. Stronę główną nowego użytkownika, który nie uzupełnił szczegółowych informacji profilu prezentuje poniższy zrzut ekranu.



Po kliknięciu na odnośnik o uzupełnieniu danych użytkownik zostaje odesłany do panelu zarządzania swoim profilem gdzie może sprecyzować swoje cechy w postaci wagi, wzrostu, wieku, płci. Może też podać swoje dane w postaci imienia i nazwiska, numeru telefonu czy zmienić swoje obecne hasło.

The screenshot shows a web interface for updating a user profile. At the top, there is a purple navigation bar with three tabs: 'Shipping', 'Edit Your Profile' (which is active), and 'Payment'. Below the navigation bar, the main heading reads 'Please insert first all your information'. The form consists of two columns of input fields. The left column contains: 'First Name' (with the value 'Łukas'), 'Last Name' (with the value 'Zach'), 'Email Address' (with the value 'mail.contactme@gmail.com'), 'Phone Number' (with the value '6564564'), 'Username' (with the value 'User'), and 'Gender' (a dropdown menu currently showing 'Male'). The right column contains: 'Insert current Password to Update' (a password field with masked characters), 'New Password' (a password field with masked characters), 'Confirm password' (a password field with masked characters), 'Your height' (with the value '180'), 'Your weight' (with the value '75'), and 'Birth Date' (with the value '02.02.1991'). At the bottom center of the form is a purple button labeled 'Update My Info'. The footer of the page is a grey bar containing the text 'IUI - Zachariasz - Zakrzewski - 11D22B - © 2020 Copyright'.

Po sprecyzowaniu danych, w momencie aktualizacji profilu następuje przeliczenie wszystkich wartości, wyznaczenie wskaźnika masy ciała, zapotrzebowania kalorycznego itd. W tym momencie po przejściu na stronę główną użytkownik uzyska w pełni spersonalizowane propozycje posiłków. Zrzut ekranu umieszczony poniżej prezentuje zaproponowane posiłki kolejno: według pory dnia, dobrania posiłku według kalorii na podstawie procentowego udziału w dziennym zapotrzebowaniu oraz ilości tłuszczów w posiłku względem BMI (nadwaga, waga prawidłowa, niedowaga).

Odświeżanie propozycji zostało ustawione na interwał 5 sekund.

Best dishes for your daily caloric needs

Breakfast

22. Tosty Z Serem



Weight: 100 (g)

Energy: 501 (kcal)

Price: 8 (PLN)

Fat: 18 (%)

[VIEW](#)

Dinner

Roladki Drobiowe



Weight: 232 (g)

Energy: 479 (kcal)

Price: 27 (PLN)

Fat: 18 (%)

[VIEW](#)

Lunch

Fit Kulki



Weight: 175 (g)

Energy: 301 (kcal)

Price: 19.9 (PLN)

Fat: 7 (%)

[VIEW](#)

Supper

Bataty Z Chrupiącą



Weight: 158 (g)

Energy: 450 (kcal)

Price: 16 (PLN)

Fat: 12 (%)

[VIEW](#)

Best dishes for your body fat condition improvement

Breakfast

22. Tosty Z Serem



Weight: 100 (g)

Energy: 501 (kcal)

Price: 8 (PLN)

Fat: 18 (%)

[VIEW](#)

Dinner

Makaron Z Passatą



Weight: 258 (g)

Energy: 488 (kcal)

Price: 19.9 (PLN)

Fat: 14 (%)

[VIEW](#)

Lunch

Czekoladowe Muffinki



Weight: 348 (g)

Energy: 528 (kcal)

Price: 20 (PLN)

Fat: 23 (%)

[VIEW](#)

Supper

Salatka Z Kurczakiem



Weight: 170 (g)

Energy: 539 (kcal)

Price: 17 (PLN)

Fat: 9 (%)

[VIEW](#)


Na samym dole strony wyświetlana jest tabela z podsumowaniem dla użytkownika.

Łukas, here is your health stats summary	
BMI Indicator	23
Rating your body condition	You have good weight for yor body
Weight	75 kg
Height	180 cm
Daily Energy Needed	1734.74 Kcal
Gender	Male

Użytkownik może w końcu wybrać dany posiłek klikając na przykład w jego zdjęcie, wówczas zostanie przeniesiony do kart produktu, gdzie może wybrać ilość do zamówienia oraz dodać produkty do koszyka. Dla przykładu kliknięto w losowo wybrany produkt, kartę posiłku prezentuje poniższy zrzut ekranu.

Food Order Store View All Food My Shopping Cart My Profile Logout Me

Go Back



Wrapy Z Jajkiem

Category	Breakfast
Energy	530 kcal
Weight	194 g
Total Energy	1028.2 kcal

Quantity

1

Add to cart

Description

Rodzaj Kanapki Wykonanej Z Okrągłego Płaskiego Chlebka (Np. Meksykańska Tortilla, Arabska Pita Lub Ormiański Lawasz) Z Zawiniętym W Środku Nadzieniem. Wrap Różni Się Od Kanapki Tym, Że Całkowicie Spowija Nadzienie. Natomiast Kanapka Składa Się Z Wyraźnie Widocznych Dwoch Warstw – Górnej I Dolnej. We Wrapie Nadzienie Z Reguły Składa Się Z Pokrojonego W Plastry Mięsa, Drobiu Lub Ryby Na Zimno Wraz Z Rozdrobnioną Sałatą, Pomidorami Pokrojonymi W Kostkę, Smażonymi Pieczarkami, Boczkem, Cebulą, Serem I Sosem. Wrapy Z Pomidorem Krojonym W Plastry, Cebulą Krojoną W Piórka, Roztopionymi Plasterkami Sera Żółtego, Sałatą, Majonezem, Ketchupem I Szybką. Wrapy Zapiekane W Opiekaczu. Zestaw Zawiera 2 Wrapy.

Po dodaniu kilku produktów do koszyka można przejść do jego wyświetlenia oraz złożenia zamówienia. Podczas podsumowania istnieje możliwość edycji ilości konkretnego produktu dla zamówienia.

Food Order Store View All Food My Shopping Cart My Profile Logout Me

Continue Shopping

Check out!

Item	Cost	Quantity	Action	
Wrapy Z Jajkiem	16 PLN	<input type="text" value="4"/>	Delete	Update
Racuchy	12 PLN	<input type="text" value="3"/>	Delete	Update
Roladki Drobiowe	27 PLN	<input type="text" value="5"/>	Delete	Update
Total	235 PLN	12		

IUI - Zachariasz - Zakrzewski - 11D22B - © 2020 Copyright

Zanim jednak zamówienie zostanie potwierdzone należy uzupełnić domyślne dane do wysyłki oraz płatności – użytkownik może również dodać domyślne adresy dostawy i wysyłki z poziomu panelu zarządzania kontem, wówczas podczas składania zamówienia adres oraz metoda płatności zostanie automatycznie uzupełniona na etapie przeglądu zamówienia. Wybór z pośród dodanych adresów i płatności pokazano poniżej.

Manage your profile

Shipping

Edit Your Profile

Payment

List of Shipping Addresses

Add(Update) Shipping Address

Default	Shipping Address	Operations	
<input checked="" type="radio"/>	Studencka, Kielce, 33-33	Edit	Remove
<input type="radio"/>	Krakowska, Kielce, 22-333	Edit	Remove

Save

IUI - Zachariasz - Zakrzewski - 11D22B - © 2020 Copyright

Natomiast uzupełnienie adresu dostawy oraz metody płatności z poziomu zarządzania kontem użytkownika prezentuje się następująco.

[Shipping](#) [Edit Your Profile](#) [Payment](#)

[List of Credit Cards](#) [Add \(Update\) Credit Card](#)

* Give a name for your card:

Moja Karta Do Jedzonka

* Name

ING

* Street Address

Uliczna

* City

Kielce

* Zipcode

33-333

* Specify your card





Discover

* Card Number

32343

CV Code

121



Save All

[Shipping](#) [Edit Your Profile](#) [Payment](#)

[List of Shipping Addresses](#) [Add\(Update\) Shipping Address](#)

* Name

Mój Adres

* Home Number

12

* Apartment Number

3

* Street

Studencka

* City

Kielce

* Zipcode

33-33

Save All

Wracając do utworzonego zamówienia, można teraz wykorzystać dodane i ustawione jako domyślne, adres dostawy oraz metodę płatności. Wystarczy kliknąć **Checkout** nad tabelą koszyka, wówczas użytkownik zostanie przekierowany do komponentu ostatecznego podsumowania i potwierdzenia zamówienia, który przedstawiono poniżej.

Food Order Store View All Food My Shopping Cart My Profile Logout Me

Finish your order

1 Check Your Address

2 Check Your Payment Information

3 Food Review

Item	Cost	Quantity
Wrapy Z Jajkiem	16 PLN	4
Racuchy	12 PLN	3
Roladki Drobiowe	27 PLN	5
Total	235 PLN	12

Back

Confirm Your Order

IUI - Zachariasz - Zakrzewski - 11D22B - © 2020 Copyright

Food Order Store View All Food My Shopping Cart My Profile Logout Me

Order Summary

Thank you, your order has been placed.
The estimated delivery Time is 02:45:40

Continue Shopping

IUI - Zachariasz - Zakrzewski - 11D22B - © 2020 Copyright

Po kliknięciu **Confirm your order** pojawi się informacja o czasie dostawy zamówienia.

Order Confirmation - 462

Odebrane x

do mnie

01:24 (0 minut temu)

angielski

>

polski

Przetłumacz wiadomość

Wyłącz dla następującego języka: angielski

Thank you for shopping from our Food Order Store.

Order Detail Number: 462

Billing Details	Payment Information	Shipping Address
ING Kielce Uliczna Kielce 33-333	Moja Karta Do Jedzonka 32343	Mój Adres Studencka Kielce 33-33 12 3

Your Order summary

Food Name	Price (PLN)	Quantity	Total (PLN)
Wrapy Z Jajkiem	16.0	4	64.00
Racuchy	12.0	3	36.00
Roladki Drobiowe	27.0	5	135.00
		Subtotal (PLN)	235.00
		Tax 8% (PLN)	18,80
		Total (PLN)	253,80

Po złożeniu zamówienia równocześnie wysłany zostanie mail z podsumowaniem.