

# System diet

Andrzej Krawczuk  
Konrad Budzyński  
Maciej Adamski

Bazy Danych – Projekt

12 Czerwca 2019

- 1 Cel projektu
- 2 Cechy funkcjonalne i нефункционалне
- 3 Baza Danych
- 4 Algorytm
- 5 Realizacja Aplikacji
- 6 Bibliografia

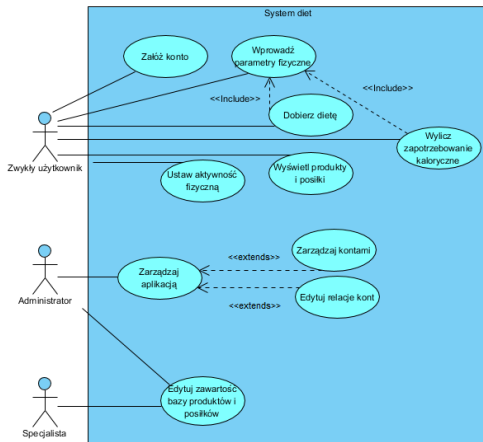
# Cel projektu



# Cechy funkcjonalne

- Zakładanie, usuwanie i konfiguracja konta
- Wprowadzanie i edycja danych
- Obliczanie zapotrzebowania na składniki odżywcze
- Dobieranie diety zależnie od zapotrzebowania
- Wyświetlanie bazy danych produktów i posiłków
- Dostęp do spersonalizowanej diety

# Digram Przypadków Użycia



Rysunek: Diagram przypadków użycia

# Cechy нефункционалне

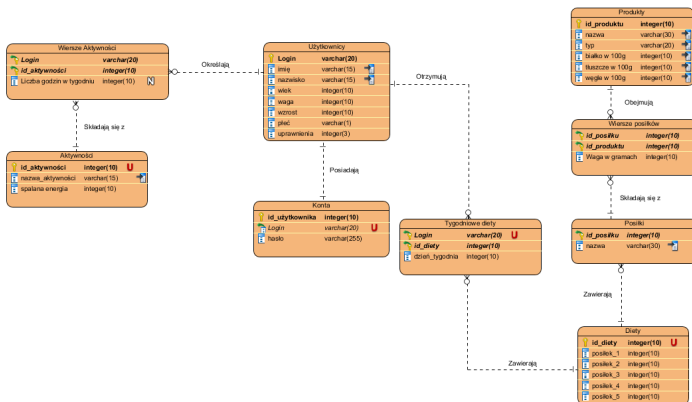
- Tryb pracy: graficzny
- Dostęp do pracy: siecowy
- System bazodanowy: MySQL
- Platforma systemowa: Windows10
- Język programowania: Python3
- Środowisko programityczne: Pycharm Community Edition
- System kontroli wersji: Git

# Przykładowy wygląd



Rysunek: Widok podstawowego panelu aplikacji

# Architektura bazy danych



Rysunek: Model logiczno-fizyczno-konceptualny



# Algorytm dobierający diety

```
def find_wfc_by_id(id):  
    cursor.execute(  
        "SELECT bialko_w_100g, tluszcz_w_100g, wegla_w_100g FROM produkty WHERE id_produktu={}".format(id)  
    )  
    result = cursor.fetchall()  
    try:  
        return result  
    except Exception as e:  
        print(e)  
        pass  
  
def return_calories_from_ingredients(data):  
    return data[0][0]*4 + data[0][1]*9 + data[0][2]*4  
  
def count_calories_in_product_by_productid(id):  
    makros = find_wfc_by_id(id)  
    return return_calories_from_ingredients(makros)  
  
def count_calories_in_meal_by_mealid(id):  
    ingredients = find_row_meal_by_id(id)  
    sum = 0  
    for ingredient in ingredients:  
        sum = sum + count_calories_in_product_by_productid(ingredient)  
    return sum
```

Rysunek: Skrypt w języku Python3 realizujący algorytm

# Backend

## Python3

Backend aplikacji napisany został w języku Python3, bez używania dodatkowych API. Połączenie do bazy danych wykorzystywało bibliotekę *MySQLDB*, a połączenie ze Frontendem wykorzystowało biblioteki takie jak *jinja2* oraz *wtfforms3*

```
@app.route('/hardsingle')
@app.route('/hardsingle/<string:db_name>/')
def hardsingle(db_name_='wiersze_aktywnosci'):
    if db_name == 'wiersze_aktywnosci':
        data, row_headers = dbconnect.select_all_test_for_login(db_name, session["username"])
        # row_headers = [x[0] for x in data.description]
        # data, columns = get_data(db_name)
        for item, i in zip(data, range(len(data))):
            newitem = list(item)
            newitem[1] = str(dbconnect.find_activity_name(item[1]))
            data[i] = newitem
            row_headers[1] = 'Nazwa_aktywnosci'
    else:
        name, surname = dbconnect.find_user_by_login(session["username"])[1:3]
        data, row_headers = dbconnect.select_diet_view(name, surname)
    data, row_headers = get_data.prepare_correct_format(data, row_headers)
    return jsonify(data_=data, col_=row_headers)

class RegistrationForm(Form):
```

# Frontend

## HTML, CSS, Javascript

Frontend aplikacji jest typową aplikacją webową realizowaną przy użyciu HTML5, CSS'ów oraz Javascriptu. Wykorzystanie ich pozwoliło stworzyć interaktywną oraz przejrzystą aplikację.

```
</style>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script type='text/javascript'>
$.noConflict();

function getData(elementId, db_name) {
    jQuery.getJSON('/hardsingls/'+db_name,
    function(json) {
        $(elementId).bootstrapTable({
            data: json.date,
            columns: json.col,
        });
    });
}

function getNewData(elementId, db_name) {
    jQuery.getJSON('/hardsingls/'+db_name,
    function(json) {
        $(elementId).bootstrapTable('load', {
            data: json.date,
            columns: json.col,
        });
    });
}
```

# Bibliografia

- 1 <http://wyborcza.pl/TylkoZdrowie/7,137474,22053386,dieta-ktora-pomoze-na-wszystko-warzywa-i-owoce-do-zadan.html?disableRedirects=true>
- 2 [www.kcalmar.com](http://www.kcalmar.com)
- 3 <https://docs.python.org/3/>
- 4 <http://flask.pocoo.org/docs/1.0/>

Dziękujemy za uwagę.