

Politechnika Poznańska  
Wydział Elektryczny  
Instytut Automatyki, Robotyki i Inżynierii Informatycznej

Praca dyplomowa inżynierska

# **DIALOG - ELEKTRONICZNY DZIENNICZEK DIABETYKA**

Krzysztof Figiel

Promotor  
dr inż. Izabela Janicka-Lipska

Poznań, 2018 r.



Tutaj przychodzi karta pracy dyplomowej;  
oryginał wstawiamy do wersji dla archiwum PP, w pozostałych kopiach wstawiamy ksero.



## Streszczenie

Za cel pracy postawiono stworzenie elektronicznego dziennika przeznaczonego dla osób chorych na cukrzycę. Dodatkowym udogodnieniem oferowanym przez aplikację jest możliwość uzyskania informacji wspomagających leczenie cukrzycy poprzez korzystanie z dostępnych kalkulatorów – BMI, wymienników węglowodanowych i białkowo-tłuszczowych. Osoba prowadząca dziennik ma dostęp do zestawień zaprezentowanych w postaci wykresów kołowych i słupkowych oraz tabeli podsumowującej wyniki. Dodatkowym atutem jest responsywność aplikacji, czyli jej dostępność zarówno na urządzeniach mobilnych jak i desktopowych.

## Abstract

The main aim of this graduation work is to create electronic diabetic register for people who suffer from diabetes. Additional support offered by the application is the ability to obtain utility for user information which are supporting diabetes treatment by using the available calculators – BMI (*Body Mass Index*), carbohydrate and protein-fat exchangers. User has permanent access to bar and pie charts as well as results summary table. An additional advantage is the responsiveness of the application – it is available on both mobile and desktop devices.



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Wprowadzenie . . . . .	1
1.2	Cel projektu . . . . .	1
1.3	Różnice pomiędzy istniejącymi aplikacjami . . . . .	1
<b>2</b>	<b>Technologie oraz wymagania</b>	<b>3</b>
2.1	Wprowadzenie . . . . .	3
2.2	Technologie . . . . .	3
2.2.1	Framework Angular 2.0.0 . . . . .	3
2.2.2	TypeScript 2.0.3 . . . . .	3
2.2.3	HTML5 . . . . .	4
2.2.4	CSS3 . . . . .	4
2.2.5	LESS 2.7.2 . . . . .	4
2.2.6	Bootstrap 3.3.7 . . . . .	5
2.2.7	Sequelize 4.2.1 . . . . .	5
2.2.8	jQuery 3.1.1 . . . . .	5
2.2.9	Node.js 3.1.1 . . . . .	5
2.3	Narzędzia . . . . .	5
2.3.1	Visual Studio Code 1.17 . . . . .	5
2.3.2	MySQL Workbench 6.3.8 . . . . .	6
2.3.3	Git CLI . . . . .	6
2.3.4	Inkscape 0.91 . . . . .	6
2.3.5	Auth0 . . . . .	6
2.4	Wymagania funkcjonalne . . . . .	6
2.4.1	SZBD - System Zarządzania Bazą Danych . . . . .	6
2.4.2	Administrator . . . . .	7
2.4.3	Użytkownik zarejestrowany . . . . .	7
2.4.4	Gość . . . . .	8
2.5	Wymagania niefunkcjonalne . . . . .	8
2.6	Analiza podobnych aplikacji . . . . .	8
2.6.1	Cukiereczek . . . . .	8
2.6.2	mySugr: Dzienniczek cukrzyka . . . . .	9
<b>3</b>	<b>Procesy zachodzące w systemie</b>	<b>11</b>
3.1	Rejestracja . . . . .	11
3.2	Logowanie . . . . .	12
3.3	Wprowadzanie i edycja informacji o profilu . . . . .	12
3.4	Wprowadzanie i edycja danych medycznych . . . . .	13
3.5	Dodawanie nowego pomiaru . . . . .	13
3.6	Dostęp do wykresów i tabeli zestawień . . . . .	13
3.7	Obliczanie wartości BMI . . . . .	14

3.8	Obliczanie wartości wymienników węglowodanowych i białkowo-tłuszczowych na podstawie dostępnej bazy produktów . . . . .	14
3.9	Dodawanie produktów do bazy danych produktów . . . . .	15
3.10	Wysyłanie wiadomości e-mail z zapytaniem dotyczącym serwisu przy użyciu modułu do wysyłania poczty elektronicznej . . . . .	15
3.11	Diagram przypadków użycia . . . . .	15
<b>4</b>	<b>Baza danych</b>	<b>17</b>
4.1	Opis tabeli bazy danych . . . . .	17
4.1.1	Person . . . . .	17
4.1.2	Medical Data . . . . .	18
4.1.3	Glucometer . . . . .	18
4.1.4	Medical Data has Tablets . . . . .	18
4.1.5	Tablets . . . . .	19
4.1.6	Medical Data has Insulin . . . . .	19
4.1.7	Diabetes Type . . . . .	19
4.1.8	Insulin . . . . .	19
4.1.9	Measurement . . . . .	19
4.1.10	Measurement has Insulin . . . . .	20
4.1.11	Glycemia Ranges . . . . .	20
4.1.12	Products . . . . .	20
4.2	Model bazy danych . . . . .	20
<b>5</b>	<b>Panel administratora</b>	<b>25</b>
5.1	Logowanie do panelu administratora . . . . .	25
5.2	Użytkowanie panelu administratora . . . . .	26
<b>6</b>	<b>Panel użytkownika zarejestrowanego</b>	<b>29</b>
6.1	Logowanie do panelu użytkownika . . . . .	29
6.2	Użytkowanie panelu użytkownika zarejestrowanego . . . . .	30
6.2.1	Twoja glikemia . . . . .	30
6.2.2	Dodaj pomiar . . . . .	33
6.2.3	Kalkulatory . . . . .	33
6.2.4	Twój profil . . . . .	36
6.2.5	Ustawienia profilu . . . . .	38
<b>7</b>	<b>Panel użytkownika niezarejestrowanego oraz niezalogowanego</b>	<b>39</b>
7.1	Okno rejestracji użytkownika . . . . .	39
7.1.1	Rejestracja przy użyciu konta <i>Google</i> . . . . .	39
7.1.2	Rejestracja tradycyjna – podanie adresu e-mail oraz hasła . . . . .	40
<b>8</b>	<b>Bezpieczeństwo aplikacji</b>	<b>43</b>
<b>9</b>	<b>Testowanie aplikacji</b>	<b>45</b>
9.1	Debugowanie aplikacji i dostęp do informacji o jej działaniu . . . . .	45
9.2	Wydajność aplikacji . . . . .	46
<b>10</b>	<b>Zakończenie</b>	<b>49</b>
	<b>Literatura</b>	<b>51</b>
<b>A</b>	<b>Załączniki</b>	<b>53</b>



# Rozdział 1

## Wstęp

### 1.1 Wprowadzenie

Wybór tematu pracy inżynierskiej spowodowany był chęcią stworzenia aplikacji, która mogłaby wspomóc leczenie cukrzycy – choroby cywilizacyjnej XXI wieku. Z roku na rok, coraz większy odsetek ludzi dołącza do grupy osób cierpiących na tę chorobę. Głównym powodem jest najczęściej niewłaściwie prowadzona dieta, a potem brak systematyczności w prowadzeniu pomiarów. Biorąc pod uwagę fakt, iż w dzisiejszych czasach prawie każdy korzysta z Internetu, zarówno na urządzeniach desktopowych jak i mobilnych stwierdzono, że można by to wykorzystać do stworzenia aplikacji, mającej na celu ułatwienie prowadzenia pomiarów i wspomagającej kontrolę cukrzycy. Wystarczy bowiem komputer, bądź urządzenie mobilne, które przeważnie każdy ma przy sobie, aby wprowadzić wartości glikemii i uzyskać dostęp do statystyk i zestawień. Dodatkowo, przed każdym spożyciem produktu użytkownik ma możliwość sprawdzenia jego makroskładników, bądź też rozszerzenia bazy danych o kolejne artykuły spożywcze, co znacząco ułatwia kontrolę prowadzonej diety.

### 1.2 Cel projektu

Głównym celem projektu jest stworzenie spójnej bazy danych, gromadzącej wszystkie utylitarne informacje dostarczane przez użytkowników oraz implementacja samego serwisu internetowego, przetwarzającego te informacje i prezentującego je w prostej i czytelnej formie. Główne funkcjonalności dostępne w aplikacji obejmują rejestrację i logowanie do systemu, dodawanie pomiarów, dodawanie i aktualizację danych dotyczących użytkownika i jego choroby, dostęp do kalkulatorów BMI, wymienników węglowodanowych czy węglowodanowo-tłuszczowych wraz z dostępną bazą produktów, czy dostęp do karty zestawień prezentującej zgromadzone informacje. Aplikacja kierowana jest do osób w dowolnym wieku, przez co jej interfejs został stworzony tak, aby jego obsługa nie wymagała specjalnych umiejętności obsługi komputera, czy urządzeń mobilnych.

### 1.3 Różnice pomiędzy istniejącymi aplikacjami

Główną zaletą projektu jest jego dostępność. Większość aplikacji dostępnych na rynku ograniczona jest do jednego systemu operacyjnego. Tutaj celem było stworzenie aplikacji dostępnej z poziomu przeglądarki internetowej, co znacznie poszerza jej zakres kompatybilności.



## Rozdział 2

# Technologie oraz wymagania

### 2.1 Wprowadzenie

W poniższym rozdziale opisano najważniejsze technologie użyte podczas implementowania projektu. W kolejnych sekcjach przedstawiono zestaw wymagań funkcjonalnych z podziałem na biorących udział aktorów oraz wymagania niefunkcjonalne.

### 2.2 Technologie

#### 2.2.1 Framework Angular 2.0.0

*Angular* jest platformą ułatwiającą tworzenie aplikacji webowych, która jest stale rozwijana przez firmę *Google*. Framework ten jest całkowicie darmowy, również do zastosowań komercyjnych. Szczególnym udogodnieniem oferowanym przez *Angulara* jest możliwość rozdzielenia logiki aplikacji od warstwy prezentacji, co znacznie ułatwia nawigowanie po komponentach projektu czy szybką identyfikację błędów. *Backend* aplikacji nie zajmuje się typowym generowaniem kodu HTML (*HyperText Markup Language*), a dostarczaniem surowych danych, np. przy pomocy REST (*Representational State Transfer*). *Angular 2* znacznie wspomaga i przyspiesza tworzenie warstwy klienckiej aplikacji. *Framework* ten służy do tworzenia stron typu SPA (*Single Page Application*), co oznacza, że żądanie, które wykonywane jest przy uruchomieniu witryny oraz pierwsze wejście do aplikacji internetowej powinno zapewnić dostarczenie wszystkich wymaganych plików arkusza stylów oraz całego szablonu. Dzięki takiemu rozwiązaniu możliwe jest dynamiczne przeładowywanie zawartości strony bez potrzeby jej odświeżania. Warto również dodać, że pozostałe dane wczytywane są asynchronicznie z *backendu* w zależności od użytego kontekstu. [Dee16]

#### 2.2.2 TypeScript 2.0.3

*TypeScript* jest otwartym językiem programowania stworzonym przez firmę *Microsoft*. Każdy może wziąć udział w jego tworzeniu i stałym rozwijaniu. Jest on rozszerzeniem funkcjonalności języka *JavaScript*. Pozwala on na tworzenie silnie typowanego, obiektowego kodu, który jest następnie kompilowany do czystego kodu *JavaScript*. Dzięki takiemu rozwiązaniu może on być interpretowany przez przeglądarki internetowe, bądź też innych klientów, takich jak np. *Node.js*. Kolejną ważną cechą jest możliwość manipulowania elementami DOM (*Document Object Model*). *TypeScript* udostępnia programistom zarówno elementy pochodzące z języka *ECMAScript*, jak i inne, dodatkowe możliwości, takie jak interfejsy, typy wyliczeniowe, moduły czy opcjonalne parametry funkcji. [Nan14]

### 2.2.3 HTML5

HTML5 (*HyperText Markup Language*) jest językiem wykorzystywanym do tworzenia i prezentacji stron internetowych. Jest to najnowsza wersja języka HTML, następcą HTML4. HTML5 pozwala na bardziej precyzyjne opisywanie zawartości strony. Umożliwia komunikację z serwerem przy wykorzystaniu nowoczesnych metod. Każda strona dostępna w Internecie skonstruowana jest zgodnie ze wzorcem HTML. Dzięki takiemu rozwiązaniu każda przeglądarka może ją odpowiednio obsłużyć, zinterpretować i wyświetlić konkretne wyniki na ekranie. W kodzie HTML umieścić można wszelkiego rodzaju elementy takie jak pola tekstowe, tabele, grafiki czy nawet wideo. Dzięki temu instalowanie dodatkowych wtyczek w przeglądarkach staje się niekonieczne. HTML5 jest dodatkowo zoptymalizowany pod kątem smartfonów i tabletów, co czyni stworzone w nim aplikacje bardziej dostępnymi. Kolejną ważną cechą HTML5 jest możliwość pracy *offline*. Pozwala to użytkownikom na korzystanie z aplikacji internetowych nawet, gdy nie posiadają oni dostępu do Internetu. Język HTML5 podobnie jak sieć cały czas ewoluuje, by sprostać potrzebom jego użytkowników. Jest to standard otwarty. Jedyne co potrzeba do jego użytkowania to nowoczesna przeglądarka internetowa, która jest z reguły dostępna za darmo. [Car17]

### 2.2.4 CSS3

CSS3 (*Cascading Style Sheets*) jest to kolejna wersja kaskadowych arkuszy stylów, wprowadzająca szereg udogodnień i rozszerzająca możliwości interakcji użytkownika ze stroną internetową. W porównaniu do poprzednich wersji, zmiany w CSS3 obejmują między innymi zastosowanie animowanych elementów czy wszelkiego rodzaju efektów graficznych, takich jak gradienty oraz cienie. CSS3 jest kompatybilny ze wszystkimi stylami objętymi standardem CSS2. Nie ma zatem potrzeby modyfikowania starszych stron przy przejściu na wyższą wersję kaskadowych arkuszy stylów. Trzecia wersja CSS zyskała również modułową budowę – specyfikacja zostaje podzielona na wiele różnych dokumentów, dzięki czemu rozwój odrębnych modułów aplikacji odbywa się odrębnie, a kod jest czysty i uporządkowany. Style CSS dodawane są do elementów na podstawie ich pozycji w drzewie dokumentu (*Document Tree*). CSS jest w pełni kompatybilny z językiem HTML, co oznacza, że HTML strukturyzuje treść strony, natomiast CSS formatuje ją w odpowiedni sposób. CSS3 pozwala na całkowitą kontrolę układu graficznego dokumentów z poziomu tylko jednego arkusza stylów. Ponadto, programista ma bardziej precyzyjną kontrolę nad całym układem graficznym. [Fra15]

### 2.2.5 LESS 2.7.2

LESS (*Leaner CSS*) to tzw. preprocesor języka CSS. Oznacza to, że kod napisany w LESS jest w późniejszym czasie kompilowany do wynikowego pliku CSS. LESS pozwala na tworzenie zmiennych, funkcji oraz domieszek, co w czystym CSS nie jest możliwe. Funkcjonalność ta sprawia, że pliki CSS stają się dużo łatwiejsze w utrzymaniu i użytkowaniu oraz o wiele bardziej rozszerzalne. LESS może być używany zarówno po stronie klienta (przeglądarka internetowa), jak i po stronie serwera, np. za pomocą *Node.js*, czy *Rhino*. Zmienne w LESS pozwalają na zdefiniowanie wartości w jednym miejscu, a następnie ich wykorzystanie w obrębie arkusza stylów. Dzięki temu istnieje możliwość dokonywania zmian globalnie w wielu elementach kodu jednocześnie. Dodatkowo LESS wprowadza tzw. domieszki, które pozwalają na zawarcie wszystkich właściwości jednej klasy w innej klasie. Odbywa się to poprzez załączenie nazwy klasy jako jednej z właściwości innej klasy. Ważnym atutem domieszek jest fakt, że mogą one zachowywać się jak funkcje i pobierać argumenty. Ponadto LESS oferuje możliwość zagnieżdżania reguł – zamiast tworzenia dłu-

gich nazw selektorów w celu określenia dziedziczenia można zagnieżdżać jedne selektory wewnątrz drugich. [Les]

### 2.2.6 Bootstrap 3.3.7

*Bootstrap* jest *frameworkiem* CSS, który obecnie rozwijany jest przez programistów firmy *Twitter*. Zawiera on zestaw narzędzi, które ułatwiają tworzenie graficznych interfejsów stron internetowych. *Bootstrap* opiera się na gotowych klasach HTML, CSS oraz *JavaScript*, przez co tworzenie przycisków, formularzy, wykresów czy menu nawigacyjnych staje się znacznie prostsze. [Rah14]

### 2.2.7 Sequelize 4.2.1

*Sequelize* jest opartym na *Node.js* ORM-em (*Object/Relational Mapper*), który ułatwia zarządzanie połączeniem z bazą danych. Zapewnia on łatwy dostęp do środowisk bazodanowych, takich jak *PostgreSQL*, *MySQL*, *SQLite*, czy *MSSQL* (*Microsoft SQL Server*). Posiada niesamowicie silny mechanizm migracji, mogący przekształcić istniejący schemat bazy danych w jego nową, różniącą się składniowo wersję. Zapewnia również mechanizmy synchronizacji z bazą danych, które mogą odzwierciedlać całą jej strukturę na podstawie określonego modelu danych. [Seq]

### 2.2.8 jQuery 3.1.1

*jQuery* to lekka biblioteka dla języka *JavaScript*, która ma za zadanie ułatwić korzystanie z jego kodu poprzez zastosowanie technik manipulacji drzewem DOM (*Document Object Model*). Dzięki *jQuery* możliwe jest dodawanie ciekawych efektów animacji na stronie, wykonywanie zapytań AJAX (*Asynchronous JavaScript and XML*) czy dynamiczna zmiana strony. *jQuery* pozwala w łatwy i wygodny dla programisty sposób zmieniać style przypisywane elementom na stronie, rozbudowywać obsługę zdarzeń czy też przetwarzać atrybuty węzłów dokumentu. [Duc15]

### 2.2.9 Node.js 3.1.1

*Node.js* to zaawansowane środowisko uruchomieniowe służące do tworzenia aplikacji internetowych na szeroką skalę, w szczególności serwerów www opartych na języku *JavaScript*. Jest on *frameworkiem* typu *Open Source*, dlatego też może być stale rozwijany. Silnik *Node.js* stworzony jest przez firmę *Google*. Jednym z najważniejszych zadań tego środowiska jest umożliwienie tworzenia aplikacji sterowanych zdarzeniami, które wykorzystują asynchroniczny system wejścia-wyjścia. [Nod]

## 2.3 Narzędzia

### 2.3.1 Visual Studio Code 1.17

*Visual Studio Code* jest edytorem kodu źródłowego rozwijanym przez firmę *Microsoft*, dla takich systemów operacyjnych jak *Microsoft*, *Linux* czy *macOS*. Zawiera wsparcie dla debugowania aplikacji, wbudowany system kontroli wersji *Git*, podświetlanie i kolorowanie składni, mechanizm inteligentnego uzupełniania kodu, wszelkiego rodzaju *snippets* czy narzędzia do refaktoryzacji kodu źródłowego. Użytkownik aplikacji *Visual Studio Code* może w prosty sposób zmienić jej szablon, zainstalować niezbędne dodatki, ustalić własne klawisze skrótu oraz ustawienia. Aplikacja ta jest w pełni darmowa i oparta na rozwiązaniach *Open Source*. *VS Code* obsługuje mnóstwo języków programowania, przez co praca

w jego środowisku staje się szybsza i bardziej efektywna niż w podstawowych edytorach tekstowych. [Vsc]

### 2.3.2 MySQL Workbench 6.3.8

*MySQL Workbench* jest wizualnym, zaawansowanym narzędziem służącym do projektowania baz danych, ułatwiającym ich administrowanie i utrzymanie w jednym, zintegrowanym środowisku programistycznym dla systemu baz danych *MySQL*. Program ten jest następcą takich środowisk jak *DBDesigner 4* czy też *labFORCE.net*. Funkcje *MySQL Workbench* obejmują wszystko to, czego potrzebuje osoba modelująca bazy do tworzenia skomplikowanych modeli ER (*Entity Relationship*), a także dostarczają kluczowe funkcje służące do wykonywania trudnych zadań związanych z zarządzaniem zmianami i dokumentacją. [Mys]

### 2.3.3 Git CLI

Konsola *Git CLI* (*Command Line Interface*) ułatwia uruchamianie wszystkich poleceń *Git* w folderze projektu. Jest lekkim i prostym interfejsem konsolowym, przez co staje się znacznie lepszym narzędziem, aniżeli konsola oferowana przez system *Windows*.

### 2.3.4 Inkscape 0.91

*Inkscape* to darmowe narzędzie do tworzenia i modyfikowania grafiki wektorowej. Zawiera pakiet podstawowych narzędzi oraz prosty i przejrzysty interfejs, co czyni go idealnym programem do zastosowań zarówno komercyjnych, jak i niekomercyjnych.

### 2.3.5 Auth0

Serwis *Auth0* umożliwia połączenie z dowolną aplikacją w celu zaoferowania usług uwierzytelniania użytkowników. Oferuje on metody logowania i rejestracji w serwisie za pomocą tradycyjnego loginu (adresu e-mail) i hasła, bądź mediów społecznościowych takich jak *Google*, *Facebook* czy *Twitter*. Domyślny protokół używany do integracji serwisu *Auth0* z aplikacją użytkownika i późniejszego uwierzytelniania to *OIDC* (*OpenID Connect*). Używa on prostych *tokenów* identyfikacyjnych w formacie *JSON* (*JavaScript Object Notation*). Wymiana danych odbywa się przy użyciu *JWT* (*JSON Web Token*), który zawiera wszystkie dane identyfikacyjne użytkownika. [Aut]

## 2.4 Wymagania funkcjonalne

### 2.4.1 SZBD - System Zarządzania Bazą Danych

Wymagania funkcjonalne dla Systemu Zarządzania Bazą Danych:

- możliwość monitorowania pracy systemu,
- możliwość administrowania wybranymi zasobami systemu z jednego, centralnego miejsca,
- integracja z zewnętrznym klientem *Auth0* – logowanie, rejestracja, wylogowywanie, dostęp do bazy użytkowników aplikacji,
- wymiana informacji pomiędzy bazą danych, a użytkownikiem,
- przetwarzanie wybranych informacji i prezentacja ich w postaci wykresów oraz tabeli podsumowań,

- obsługa działania kalkulatorów – BMI (*Body Mass Index*), wymienników węglowodanowych oraz wymienników białkowo-tłuszczowych.

### 2.4.2 Administrator

Wymagania funkcjonalne dla użytkownika Administrator:

- usuwanie produktów spożywczych z bazy danych,
- dostęp do informacji o użytkownikach systemu,
- dostęp do poczty e-mail modułu do wysyłania zapytań,
- dodatkowo – funkcjonalności użytkownika zarejestrowanego w celu sprawdzenia poprawności funkcjonowania systemu.

### 2.4.3 Użytkownik zarejestrowany

Wymagania funkcjonalne dla użytkownika zarejestrowanego:

- możliwość dodawania pomiarów glikemii,
- dostęp do statystyk i tabeli podsumowań dotyczących pomiarów,
- możliwość dodawania produktów spożywczych do bazy danych,
- dostęp do bazy danych produktów,
- dostęp do kalkulatora BMI oraz do kalkulatorów wymienników węglowodanowych i białkowo-tłuszczowych,
- możliwość dodawania oraz edycji danych dotyczących informacji o profilu:
  - imienia,
  - nazwiska,
  - PESEL,
  - posiadanie opiekuna,
  - adresu zamieszkania (ulica, numer domu, miasto, kod pocztowy),
  - numeru telefonu,
  - aktualnego modelu glukometru,
  - poprzedniego modelu glukometru,
  - numeru seryjnego glukometru.
- możliwość dodawania oraz edycji danych medycznych:
  - typu cukrzycy,
  - roku zachorowania,
  - rodzaju insuliny,
  - przyjmowanych leków,
  - innych leków i typów insuliny,
  - wzrostu,
  - wagi,

- poziomu HbA1c – wskaźnika poziomu hemoglobiny glikowanej, który odzwierciedla przeciętny poziom cukru we krwi.
- dostęp do informacji o stronie,
- dostęp do modułu wysyłania mailowych zapytań.

#### 2.4.4 Gość

Wymagania funkcjonalne dla użytkownika niezarejestrowanego (gościa):

- możliwość zarejestrowania się w systemie,
- możliwość przeglądania informacji o stronie i wysyłania maili z modułu do elektronicznej korespondencji.

### 2.5 Wymagania niefunkcjonalne

Wymagania niefunkcjonalne dla aplikacji:

- do prawidłowej pracy systemu niezbędna jest nowoczesna przeglądarka internetowa:
  - *Internet Explorer 9* i nowszy,
  - *Mozilla Firefox 3.0* i nowsza,
  - *Google Chrome 55.0* i nowszy.
- cała aplikacja dostępna przez przeglądarkę internetową,
- hasła podawane przy zakładaniu konta oraz logowaniu są przechowywane za pomocą funkcji skrótu *BCrypt* i gromadzone w bazie danych klienta *Auth0*,
- dane powinny być zabezpieczane przez regularne kopie zapasowe całej bazy danych,
- do prawidłowej rejestracji w systemie wymagane jest posiadanie adresu e-mail bądź konta *Google*,
- korzystanie z aplikacji wiąże się ze zgodą na przechowywanie danych osobowych przez administrację.

### 2.6 Analiza podobnych aplikacji

#### 2.6.1 Cukiereczek

*Cukiereczek* to oprogramowanie przeznaczone dla osób cierpiących na cukrzycę. Aplikacja jest dostępna lokalnie po uprzednim zainstalowaniu na systemie operacyjnym *Windows XP/Vista/7/8*. Za pomocą tego oprogramowania można kontrolować i prowadzić dzienniczek samokontroli, a także prowadzić rejestr pomiarów cukrów czy wyników badań laboratoryjnych. Program zapewnia dostęp do szczegółowych wykresów i analizy pomiarów. Aplikacja zawiera również tabelę produktów spożywczych wraz z wartościami makroskładników, które posiadają. Istotną wadą oprogramowania jest to, że jest ono ukierunkowane pod jeden rodzaj systemu operacyjnego (*Microsoft Windows*) i dostępne jedynie z poziomu urządzeń desktopowych.



### 2.6.2 mySugr: Dzienniczek cukrzyka

Jest to darmowa aplikacja dostępna na urządzenia mobilne z systemem operacyjnym *Android*. W dużym skrócie do jej podstawowych funkcji należą:

- możliwość wprowadzania szybkich pomiarów poziomu glukozy we krwi,
- spersonalizowany dzienniczek,
- wykresy cukru we krwi,
- wyzwania i cele do osiągnięcia przez użytkownika (wersja płatna),
- szacowana wartość HbA1c.

Istotną wadą aplikacji jest jednak fakt, że jest ona ograniczona jedynie do urządzeń mobilnych z wbudowanym systemem operacyjnym *Android*.

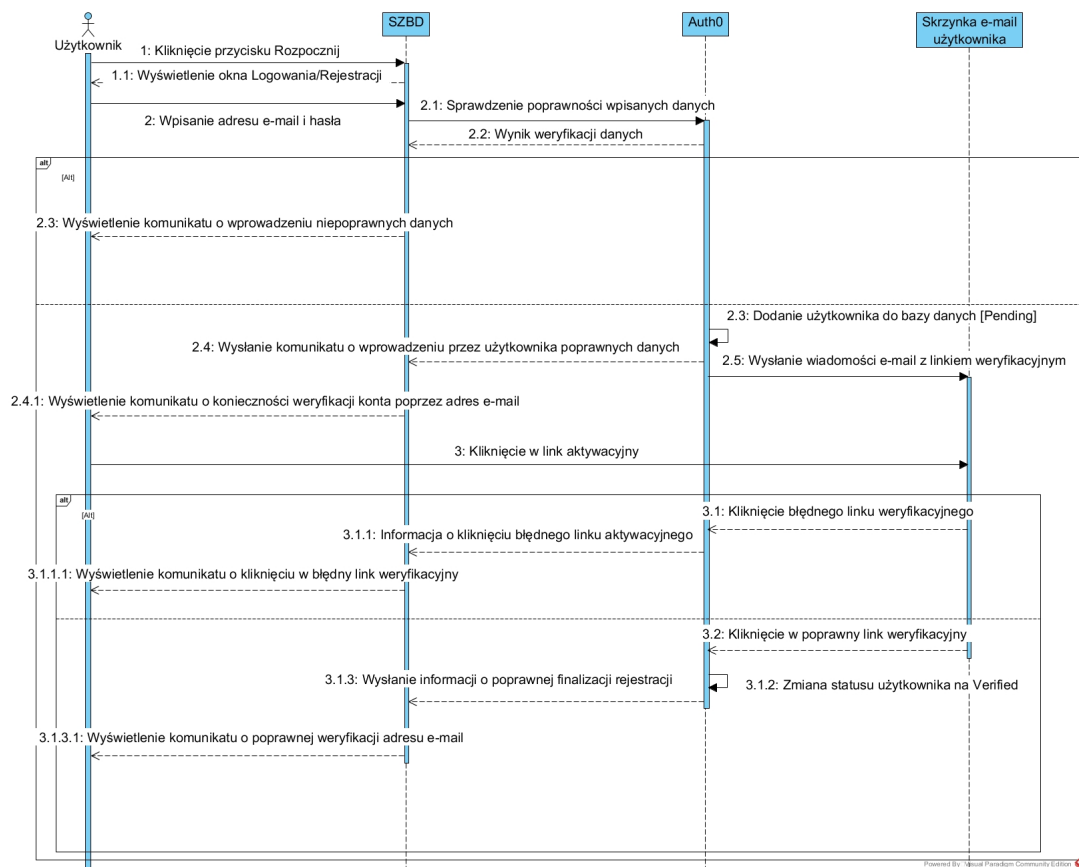


## Rozdział 3

# Procesy zachodzące w systemie

### 3.1 Rejestracja

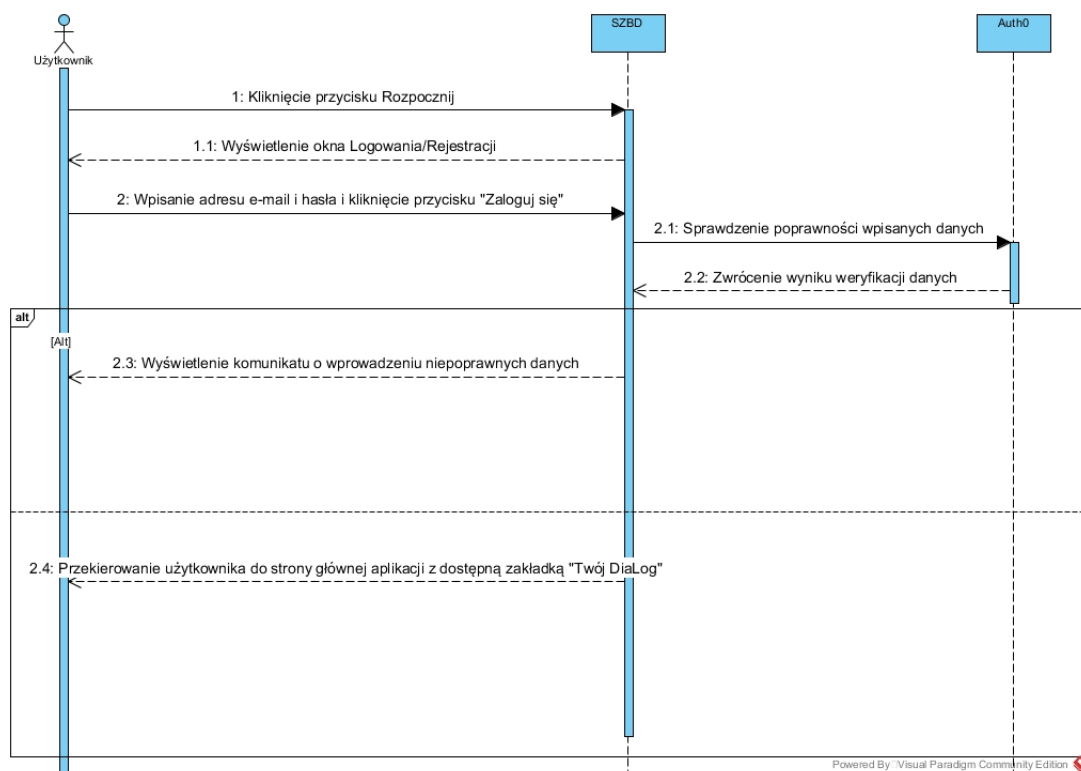
Rejestracja użytkownika w systemie realizowana jest za pomocą zewnętrznego serwisu *Auth0*.



Rysunek 3.1: Diagram sekwencji przedstawiający proces rejestracji nowego użytkownika

## 3.2 Logowanie

Logowanie użytkownika w systemie realizowane jest za pomocą zewnętrznego serwisu *Auth0*.



Rysunek 3.2: Diagram sekwencji przedstawiający proces logowania użytkownika

## 3.3 Wprowadzanie i edycja informacji o profilu

Użytkownik ma możliwość wprowadzania i późniejszej edycji danych dotyczących informacji o profilu. Odbывается to poprzez wypełnienie formularza. Użytkownik (poprzez odpowiednie wypełnienie pól formularza) określa:

- imię,
- nazwisko,
- PESEL,
- posiadanie opiekuna,
- adres zamieszkania (ulica, numer domu, miasto, kod pocztowy),
- numer telefonu,
- aktualny model glukometru,
- poprzedni modelu glukometru,
- numer seryjny glukometru.

W przypadku chęci edycji tych danych wystarczy zastąpić je w odpowiednim polu i kliknąć przycisk *Zapisz*.

### 3.4 Wprowadzanie i edycja danych medycznych

Użytkownik ma możliwość wprowadzania i późniejszej edycji danych medycznych. Odbywa się to poprzez wypełnienie formularza. Użytkownik (poprzez odpowiednie wypełnienie pól formularza) określa:

- typ cukrzycy,
- rok zachorowania,
- rodzaj insuliny,
- przyjmowane leki,
- inne leki i insuliny,
- wzrost,
- wagę,
- poziom współczynnika HbA1c.

W przypadku chęci edycji tych danych, podobnie jak w poprzednim przypadku należy zastąpić je odpowiednimi informacjami i wybrać przycisk *Zapisz*.

### 3.5 Dodawanie nowego pomiaru

Użytkownik ma możliwość dodania nowego pomiaru. Czynność ta odbywa się poprzez kliknięcie przycisku *Nowy pomiar*. Po wykonaniu tej czynności ukazuje się okno modalne z odpowiednimi polami do wypełnienia:

- data i godzina pomiaru,
- glikemia,
- moment pomiaru,
- krótka notatka.

Po wypełnieniu wszystkich pól formularza należy kliknąć przycisk *Dodaj*. Okno modalne zostaje wówczas automatycznie zamknięte, a nowo prowadzone dane zostają dodane do sekcji wykresów i tabeli podsumowań.

### 3.6 Dostęp do wykresów i tabeli zestawień

Użytkownik w każdym momencie może uzyskać dostęp do wykresów i tabeli zestawień poprzez wybranie pozycji menu *Twoja glikemia*. Wykresy oraz dane zawarte w tabeli generowane są w sposób automatyczny po dodaniu nowej wartości pomiaru.

### 3.7 Obliczanie wartości BMI

Użytkownik w dowolnym momencie może sprawdzić swoją wartość współczynnika BMI (*Body Mass Index*) poprzez wybranie z menu bocznego pozycji *Kalkulatory*, a następnie wprowadzenie swojej wagi oraz swojego wzrostu i kliknięcie przycisku *Oblicz*. W momencie wyświetlenia wyniku program podpowiada użytkownikowi jak współczynnik BMI odzworowuje się na jego aktualnej sylwetce i kondycji zdrowotnej:

- niedowaga,
- waga prawidłowa,
- nadwaga,
- otyłość I stopnia,
- otyłość II stopnia,
- otyłość III stopnia.

### 3.8 Obliczanie wartości wymienników węglowodanowych i białkowo-tłuszczowych na podstawie dostępnej bazy produktów

Użytkownik ma możliwość obliczenia współczynnika wymienników węglowodanowych oraz wymienników białkowo-tłuszczowych na podstawie dostępnej bazy produktów, która może być rozwijana przez użytkowników. W pierwszej kolejności użytkownik wybiera produkt z pola formularza z wbudowanym inteligentnym modulem wyszukiwania produktów. W następnej kolejności użytkownik wybiera wagę produktu (domyślna wartość to 100 gram). W tabeli zawartej poniżej formularza użytkownik ma dostęp do tabeli makroskładników danego produktu:

- kilokalorie produktu [kcal],
- ilość białka [g],
- ilość tłuszczu [g],
- ilość węglowodanów [g],
- ilość błonnika [g],
- wielkość porcji [g],
- wartość współczynnika wymienników węglowodanowych,
- wartość współczynnika wymienników białkowo-tłuszczowych.

Wartości zawarte w tabeli są dynamicznie zmieniane w zależności od wprowadzonej gramatury produktu.

### 3.9 Dodawanie produktów do bazy danych produktów

Użytkownik ma możliwość dodania produktu do bazy danych produktów. Odbywa się to za pomocą wypełnienia odpowiednich pól formularza:

- nazwa produktu,
- kalorie [kcal],
- białko [g],
- tłuszcz [g],
- węglowodany [g],
- błonnik [g],
- porcja [g].

Aby zatwierdzić zmiany należy kliknąć przycisk *Zapisz*. Produkt automatycznie zostanie dodany do bazy danych i w sposób dynamiczny będzie dostępny w wyszukiwarce produktów.

### 3.10 Wysłanie wiadomości e-mail z zapytaniem dotyczącym serwisu przy użyciu modułu do wysyłania poczty elektronicznej

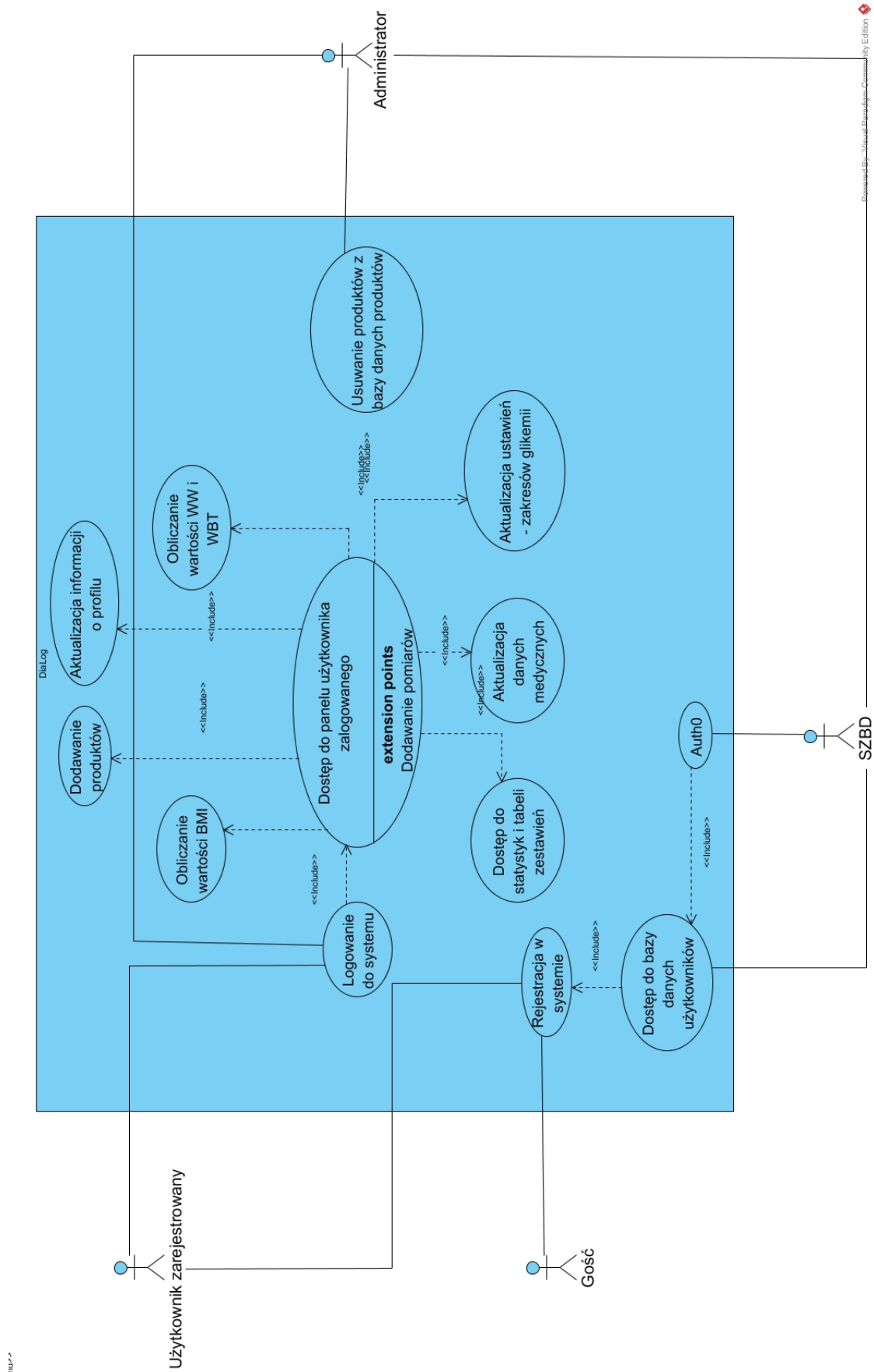
Użytkownik ma możliwość wysłania wiadomości e-mail do administracji strony poprzez wypełnienie odpowiednich pól formularza dostępnych na stronie startowej:

- imię,
- e-mail,
- temat,
- treść wiadomości.

W celu zatwierdzenia wiadomości e-mail należy kliknąć przycisk *Wyślij*.

### 3.11 Diagram przypadków użycia

W poniższej sekcji, na rysunku 3.3 zaprezentowany został diagram przypadków użycia dla każdego z aktorów.



Rysunek 3.3: Diagram przypadków użycia dla aktorów aplikacji



## Rozdział 4

# Baza danych

### 4.1 Opis tabeli bazy danych

W poniższej sekcji opisane zostaną tabele bazy danych wraz z polami w nich zawartymi. Określony zostanie typ danego pola oraz krótki opis jego przeznaczenia.

#### 4.1.1 Person

Tabela *person* – zawiera dane dotyczące użytkownika aplikacji:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *name* – pole typu *VARCHAR* przechowujące imię użytkownika,
- *surname* – pole typu *VARCHAR* przechowujące nazwisko użytkownika,
- *pesel* – pole typu *VARCHAR* przechowujące numer pesel użytkownika,
- *guardian* – pole typu *VARCHAR* mówiące o tym, czy dany użytkownik posiada opiekuna,
- *street* – pole typu *VARCHAR* przechowujące nazwę ulicy, na której mieszka użytkownik,
- *homeNumber* – pole typu *VARCHAR* przechowujące numer domu, pod którym mieszka użytkownik,
- *city* – pole typu *VARCHAR* przechowujące nazwę miasta, w którym mieszka użytkownik,
- *postalCode* – pole typu *VARCHAR* przechowujące kod pocztowy miasta, w którym mieszka użytkownik,
- *telephoneNumber* – pole typu *VARCHAR* przechowujące numer telefonu użytkownika,
- *actualGlucometer* – pole typu *INT* będące kluczem obcym z tabeli *glucometer*, zawierające *id* aktualnie posiadanego glukometru,
- *previousGlucometer* – pole typu *INT* będące kluczem obcym z tabeli *glucometer*, zawierające *id* poprzedniego glukometru,
- *serialNumber* – pole typu *VARCHAR* przechowujące numer seryjny glukometru, którego aktualnie używa użytkownik.

#### 4.1.2 Medical Data

Tabela *medical\_data* - zawiera informacje dotyczące danych medycznych użytkownika:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *yearOfIllness* – pole typu *TEXT* przechowujące informację na temat roku zachorowania użytkownika na cukrzycę,
- *otherMedicines* – pole typu *VARCHAR* przechowujące informacje na temat innych, przyjmowanych leków,
- *bmi* – pole typu *DECIMAL* przechowujące wartość współczynnika BMI (*Body Mass Index*) użytkownika,
- *height* – pole typu *INT* przechowujące informacje na temat wzrostu użytkownika w cm,
- *weight* – pole typu *VARCHAR* przechowujące informacje na temat wagi użytkownika,
- *hbValue* – pole typu *DECIMAL* przechowujące informacje na temat wartości współczynnika HbA1c użytkownika,
- *diabetesTypeId* – pole typu *INT* będące kluczem obcym z tabeli *diabetes\_type*, zawierające *id* typu cukrzycy, na którą cierpi użytkownik,
- *insulinId* – pole typu *INT* będące kluczem obcym z tabeli *insulin*, zawierające *id* typu insuliny, którą zażywa użytkownik,
- *tabetsId* – pole typu *INT* będące kluczem obcym z tabeli *tablets*, zawierające *id* leku, który przyjmuje użytkownik.

#### 4.1.3 Glucometer

Tabela *glucometer* – zawiera informacje dotyczące dostępnych glukometrów w bazie danych:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *glucometerName* – pole typu *VARCHAR* zawierające nazwę danego glukometru.

#### 4.1.4 Medical Data has Tablets

Tabela *medical\_data\_has\_tablets* - zawiera informacje dotyczące przyjmowanych przez danego użytkownika leków:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *medicalDataId* – pole typu *INT* będące kluczem obcym z tabeli *medical\_data*, zawierające *id* danych medycznych danego użytkownika,
- *tabetsId* – pole typu *INT* będące kluczem obcym z tabeli *tablets*, zawierające *id* leku, który przyjmuje użytkownik.

#### 4.1.5 Tablets

Tabela *tablets* – zawiera informacje dotyczące dostępnych leków w bazie danych:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *tabletName* – pole typu *VARCHAR* zawierające nazwę danego medykamentu.

#### 4.1.6 Medical Data has Insulin

Tabela *medical\_data\_hasinsulin* – zawiera informacje dotyczące insuliny przyjmowanej przez danego użytkownika:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *medicalDataId* – pole typu *INT* będące kluczem obcym z tabeli *medical\_data*, zawierające *id* danych medycznych danego użytkownika,
- *insulinId* – pole typu *INT* będące kluczem obcym z tabeli *insulin*, zawierające *id* typu insuliny, którą zażywa użytkownik.

#### 4.1.7 Diabetes Type

Tabela *diabetes\_type* – zawiera informacje dostępnych w bazie danych typów cukrzycy:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *typeName* – pole typu *VARCHAR* zawierające nazwę danego typu cukrzycy.

#### 4.1.8 Insulin

Tabela *insulin* – zawiera informacje dotyczące dostępnych w bazie danych rodzajów insuliny:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *insulinName* – pole typu *VARCHAR* zawierające nazwę danego typu insuliny.

#### 4.1.9 Measurement

Tabela *measurement* – zawiera informacje dotyczące pomiarów dokonywanych przez użytkowników:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *date* – pole typu *DATETIME* zawierające datę aktualnie dokonanego przez użytkownika pomiaru,
- *glicemy* – pole typu *INT* zawierające wartość glikemii, którą wprowadza użytkownik,
- *moment* – pole typu *TEXT* zawierające informację na temat momentu dokonanego pomiaru,
- *note* – pole typu *VARCHAR* zawierające notatkę dotyczącą pomiaru dokonanego przez użytkownika.

#### 4.1.10 Measurement has Insulin

Tabela *measure\_has\_person* – grupuje informacje dotyczące danego pomiaru, wykonanego przez daną osobę:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *measurementId* – pole typu *INT* będące kluczem obcym z tabeli *measurement*, zawierające *id* pomiaru dokonanego przez użytkownika,
- *personId* – pole typu *INT* będące kluczem obcym z tabeli *person*, zawierające *id* osoby, dokonującej pomiaru.

#### 4.1.11 Glycemia Ranges

Tabela *glycemia\_ranges* – zawiera informacje dotyczące zakresów glikemii użytkownika:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *hipoglicemy* – pole typu *INT* zawierające wartość hipoglikemii użytkownika,
- *hiperglicemy* – pole typu *INT* zawierające wartość hiperglikemii mierzonej przez użytkownika,
- *hiperglicemyMeal* – pole typu *INT* zawierające wartość hiperglikemii, mierzonej przez użytkownika po posiłku.

#### 4.1.12 Products

Tabela *products* – zawiera informacje dotyczące dostępnych w bazie danych produktów:

- *id* – pole typu *INT* zawierające unikalny identyfikator pola w bazie danych,
- *productName* – pole typu *VARCHAR* przechowujące nazwę danego produktu,
- *kcal* – pole typu *INT* zawierające ilość kalorii, które zawiera dany produkt,
- *protein* – pole typu *DECIMAL* zawierające ilość białka, które zawiera dany produkt,
- *fat* – pole typu *DECIMAL* zawierające ilość tłuszczu, który zawiera dany produkt,
- *carbohydrates* – pole typu *DECIMAL* zawierające ilość węglowodanów, które zawiera dany produkt,
- *fiber* – pole typu *DECIMAL* zawierające ilość błonnika, który zawiera dany produkt,
- *portion* – pole typu *DECIMAL* zawierające wielkość porcji danego produktu.

## 4.2 Model bazy danych

Rysunek 4.1 przedstawia schemat EER (*Enhanced Entity-Relationship*) modelu bazy danych użytego do gromadzenia informacji w systemie. Dla każdej tabeli został określony klucz główny, który może być używany w innych tabelach jako klucz obcy. Każde z pól w bazie danych zawiera określony typ, w zależności od tego jakie informacje ma dane pole zawierać. Tabele zawarte w bazie połączone są między sobą relacjami:

- jeden-do-jeden,

- jeden-do-wielu,
- wiele-do-wielu.

W przypadku relacji wiele-do-wielu wykorzystana została tabela grupująca klucze obce z obydwu łączonych w tę relację tabel.

Schemat EER bazy danych został wygenerowany automatycznie za pomocą programu *MySQL Workbench* w wersji 6.3 za pomocą narzędzia *Reverse Engineer*.

Zapytania do bazy danych generowane są w sposób automatyczny za pomocą ORM *Sequelize* na podstawie odwzorowanego w kodzie źródłowym modelu bazy danych. Cała baza danych w programie została wygenerowana w konwencji *Code First* – najpierw stworzony był model, a następnie na podstawie tego modelu odpowiadająca mu tabela bazy danych. Poniższy kod przedstawia przykładową implementację modelu tabeli *products*:

```
module.exports = function(sequelize, DataTypes) {
  return sequelize.define('products', {
    id: {
      type: DataTypes.INTEGER(11),
      allowNull: false,
      primaryKey: true,
      autoIncrement: true
    },
    productName: {
      type: DataTypes.STRING(45),
      allowNull: false
    },
    kcal: {
      type: DataTypes.INTEGER(11),
      allowNull: false
    },
    protein: {
      type: DataTypes.DECIMAL,
      allowNull: false
    },
    fat: {
      type: DataTypes.DECIMAL,
      allowNull: false
    },
    carbohydrates: {
      type: DataTypes.DECIMAL,
      allowNull: false
    },
    fiber: {
      type: DataTypes.DECIMAL,
      allowNull: true
    },
    portion: {
      type: DataTypes.DECIMAL,
      allowNull: false
    }
  }, {
    tableName: 'products',
    timestamps : false
  })
}
```

```
});  
};  
}
```

Początkowo eksportowany jest model przy użyciu funkcji ORM *Sequelize*. Zwracana wartość zawiera odwzorowanie tabeli produkt, na podstawie którego generowane są później automatyczne zapytania do bazy danych. Każde pole tabeli opisywane jest w następujący sposób:

```
nazwa_pola: {  
  type: typ_pola,  
  ...  
  //odpowiednie opcje i ograniczenia  
}
```

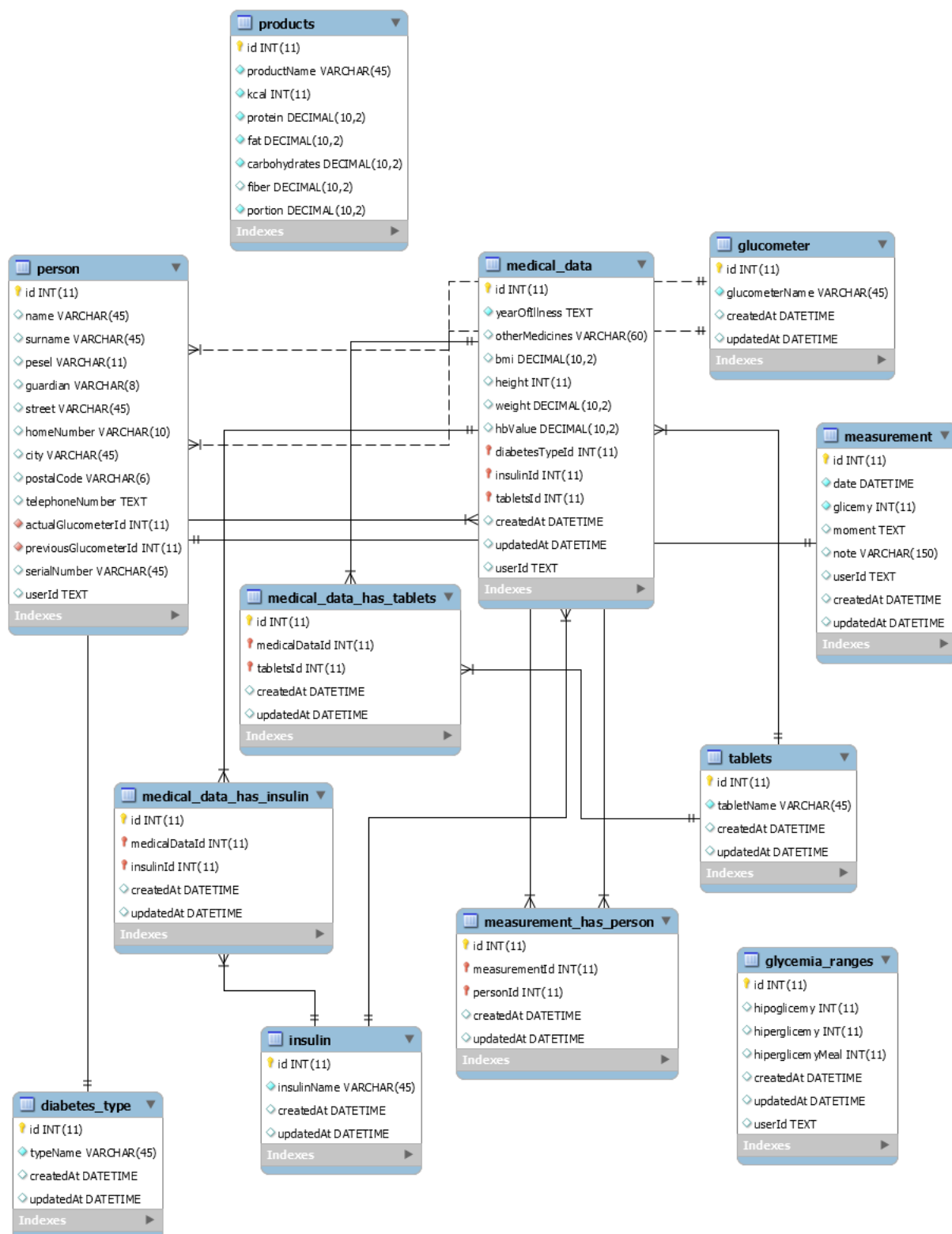
Ograniczenia jakie zostały użyte do zdefiniowania klucza głównego to:

```
allowNull: false;  
primaryKey: true;  
autoIncrement: true
```

Pierwsze z nich określa czy pole tabeli może być wartością pustą (*NULL*). Klucz główny nie może być wartością pustą, dlatego wartość ta jest ustawiona na *false*. Następna wartość określa, czy dane pole jest kluczem głównym. W tym przypadku przyjmuje ono wartość *true*. Ostatnim z pól jest określenie autoinkrementowania wartości *id*, dlatego również przyjmuje ono wartość *true*. Ostatni znacznik:

```
timestamps: false
```

określa anulowanie możliwości korzystania z automatycznych znaczników czasu tworzonych w momencie edycji *UPDATE* tabeli.



Rysunek 4.1: Diagram EER bazy danych aplikacji



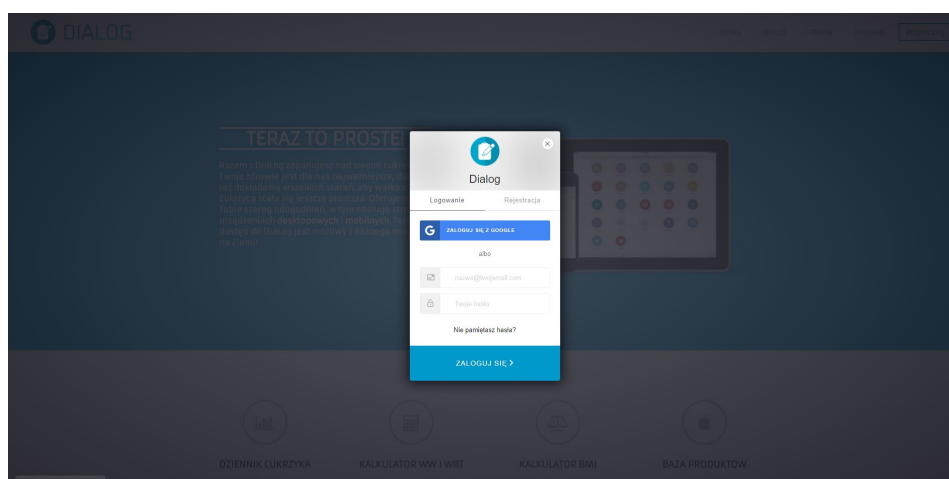


## Rozdział 5

# Panel administratora

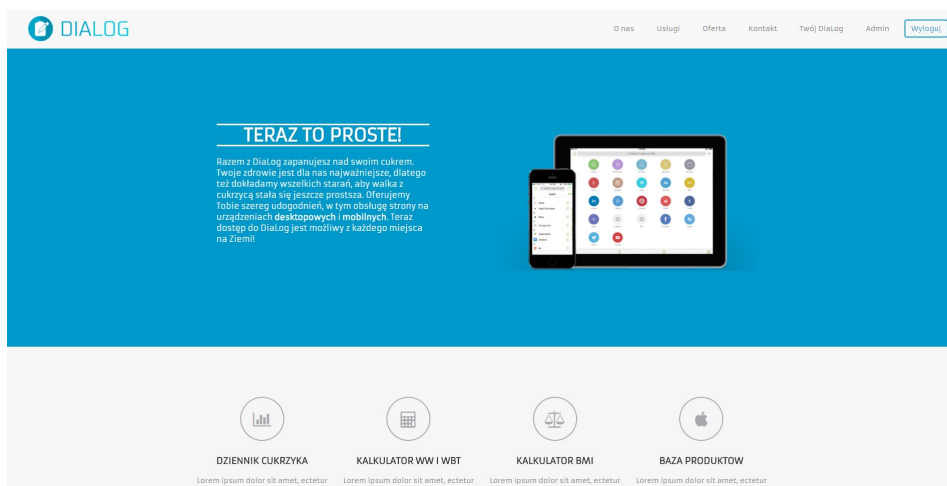
### 5.1 Logowanie do panelu administratora

W poniższej sekcji przedstawiony zostanie proces logowania do panelu administratora w aplikacji. W pierwszej kolejności użytkownik musi wybrać opcję *Rozpocznij*, dostępną z poziomu górnego menu na stronie głównej, a następnie wybrać przycisk *Logowanie z Google*. Poniższy rysunek 6.1 przedstawia okno logowania użytkownika.



Rysunek 5.1: Okno logowania użytkownika

Po wybraniu przycisku *Logowanie z Google* należy zalogować się na konto z przypisanymi uprawnieniami administracyjnymi. Pojawi się wówczas dodatkowa pozycja w menu górnym - *Admin*, co przedstawione zostało na rysunku 5.2.



Rysunek 5.2: Okno główne aplikacji z dostępną opcją panelu administratora

## 5.2 Użytkowanie panelu administratora

Po kliknięciu przycisku *Admin* użytkownik uzyskuje dostęp do Panelu Administratora. Panel ten został stworzony wyłącznie po to, aby usuwać błędnie dodane, bądź zduplikowane produkty dodane do bazy danych produktów. Jest to niezbędny element w przypadku funkcjonalności danej aplikacji rozszerzanych przez użytkowników. Dzięki takiemu rozwiązaniu Administrator w prosty sposób może kontrolować treść zawartą na stronie i w razie potrzeby usuwać ją. Panel administratora przedstawiony został na rysunku 5.3. Panel został zaprojektowany w postaci tabeli tak, aby w prosty i przejrzysty sposób zaprezentować wszystkie dostępne dane. Ponieważ baza produktów może być stale rozszerzana, a ich ilość może sięgać nawet kilkuset elementów, zastosowana została paginacja tabeli, dzieląca jej zawartość na strony zawierające po dziesięć elementów.

Nazwa produktu	Kalorie [kcal]	Białko [g]	Tłuszcz [g]	Węglowodany [g]	Błonnik [g]	Porcja [g]	Czynność
Slickers	250	20.00	15.00	60.00	10.00	100.00	Usuń
Morela	50	2.00	3.00	30.00	10.00	100.00	Usuń
Czekolada mleczna Milka	530	6.00	29.00	58.00	4.00	100.00	Usuń
Batonik MilkyWay	454	3.80	17.00	71.60	2.10	100.00	Usuń
Czekolada mleczna Goplana	530	8.00	30.00	56.00	4.00	100.00	Usuń
Czekolada mleczna Wedel	530	6.00	30.00	58.00	2.50	100.00	Usuń
Czekolada gorzka Wedel	502	8.40	32.00	38.00	13.00	100.00	Usuń
Czekolada gorzka Goplana	516	7.80	32.00	42.00	3.20	100.00	Usuń
Bułka pszenna	277	8.10	1.50	57.70	1.80	100.00	Usuń
Baton Mars	452	3.70	17.00	70.30	0.10	100.00	Usuń

Rysunek 5.3: Panel administratora aplikacji

Tabela podzielona jest na osiem kolumn:

1. *Nazwa produktu*,
2. *Kalorie [kcal]*,

3. *Białko [g]*,
4. *Tłuszcz [g]*,
5. *Węglowodany [g]*,
6. *Błonnik [g]*,
7. *Porcja [g]*,
8. *Czynność*.

W kolumnie *Czynność* przy każdym z wierszy dostępny jest przycisk *Usuń*, dzięki któremu Administrator może usunąć wybrany produkt. Usuwanie produktu odbywa się w sposób dynamiczny, co oznacza, że produkt po usunięciu od razu (bez odświeżenia strony) znika z listy dostępnych produktów i nie jest on dostępny nawet z poziomu tabeli produktów dostępnej w zakładce *Kalkulatory* z poziomu panelu użytkownika zalogowanego.

Administrator, tak jak każdy zalogowany użytkownik, ma również możliwość korzystania ze wszystkich funkcjonalności systemu przeznaczonych dla tej grupy użytkowników – zakładka *Twój DiaLog*.

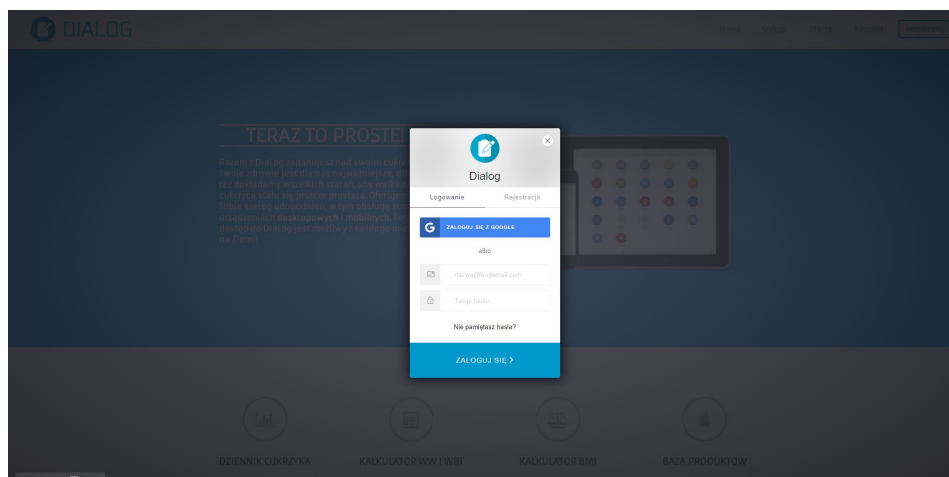


## Rozdział 6

# Panel użytkownika zarejestrowanego

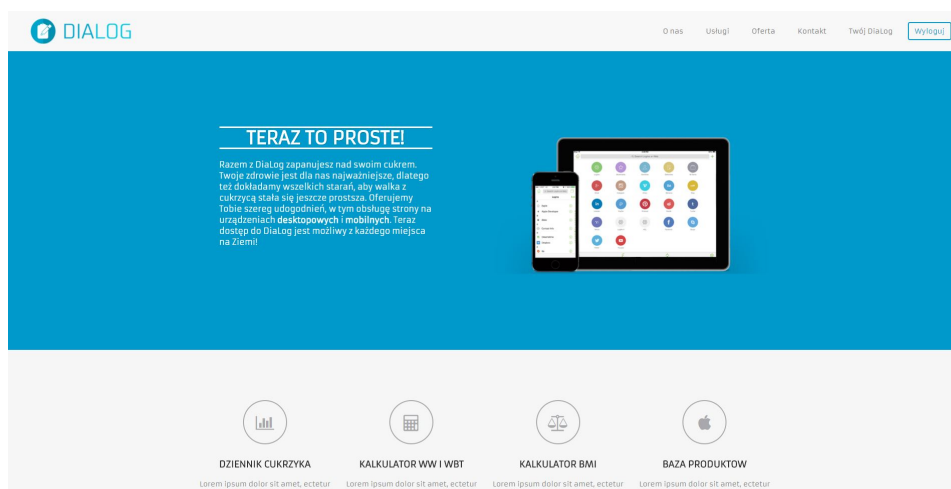
### 6.1 Logowanie do panelu użytkownika

W poniższej sekcji przedstawiony zostanie proces logowania do panelu użytkownika w aplikacji. W pierwszej kolejności użytkownik musi wybrać opcję *Rozpocznij*, dostępną z poziomu górnego menu na stronie głównej, a następnie wpisać login i hasło, bądź wybrać przycisk *Zaloguj się z Google*. Okno logowania przedstawione jest na rysunku 6.1.



Rysunek 6.1: Okno logowania użytkownika

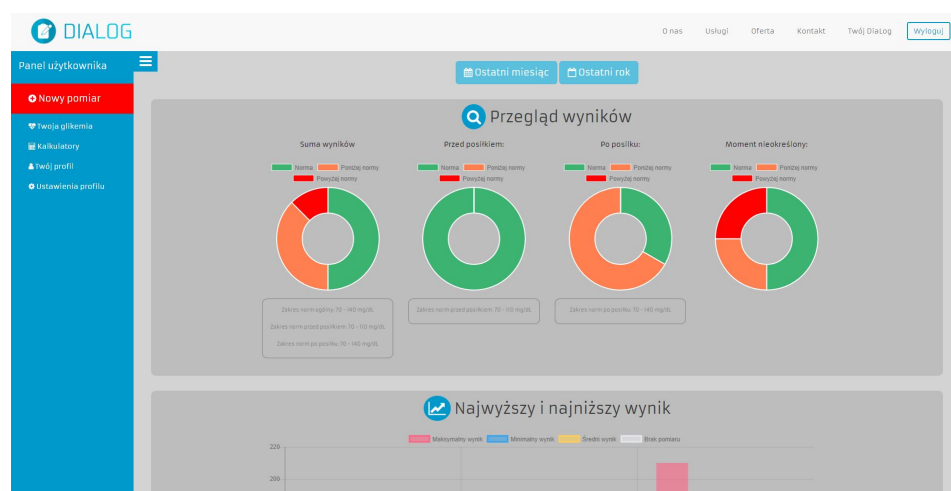
Po wpisaniu danych służących do logowania użytkownik uzyskuje dostęp do nowej pozycji w menu górnym – *Twój DiaLog*. Menu górne dostępne po zalogowaniu użytkownika zostało ukazane na rysunku 6.2.



Rysunek 6.2: Menu górne rozszerzone o dodatkową pozycję – *Twój DiaLog*, dostępne po zalogowaniu użytkownika

## 6.2 Użytkowanie panelu użytkownika zarejestrowanego

Po uzyskaniu dostępu do zakładki *Twój DiaLog* i kliknięciu jej, użytkownik uzyskuje dostęp do strony głównej panelu użytkownika zarejestrowanego będącej jednocześnie komponentem podsumowującym pomiary wprowadzone przez użytkownika. Początkowo, po rejestracji wykresy i tabela zestawień są puste, gdyż nie zawierają żadnych danych. Strona główna po kliknięciu pozycji górnego menu *Twój DiaLog* została ukazana na rysunku 6.3.



Rysunek 6.3: Strona główna dostępna po kliknięciu przycisku menu górnego *Twój DiaLog*, dostępnego po zalogowaniu użytkownika

### 6.2.1 Twoja glikemia

Zawartość dostępna z poziomu zakładki *Twoja glikemia* wyświetlana jest jako strona główna po kliknięciu zakładki *Twój DiaLog*. Zawiera ona wykresy kołowe podsumowu-

jące ilość dodanych pomiarów glikemii. Każdy z czterech wykresów kołowych dotyczy odrębnego rodzaju pomiaru. Rodzaje pomiarów glikemii rozróżniane są w następujący sposób:

- Przed posiłkiem,
- Po posiłku,
- Moment nieokreślony.

Pierwszy z wykresów jest sumą ilości pomiarów z pozostałych trzech wykresów. Ponadto, każdy z wykresów rozróżnia normy zakresu glikemii odpowiednie dla danego rodzaju pomiaru:

- Przed posiłkiem:
  - Norma ogólna: 70 - 140 mg/dL.
- Po posiłku,
  - Zakres norm przed posiłkiem: 70 - 110 mg/dL.
- Moment nieokreślony.
  - Zakres norm po posiłku: 70 - 140 mg/dL.

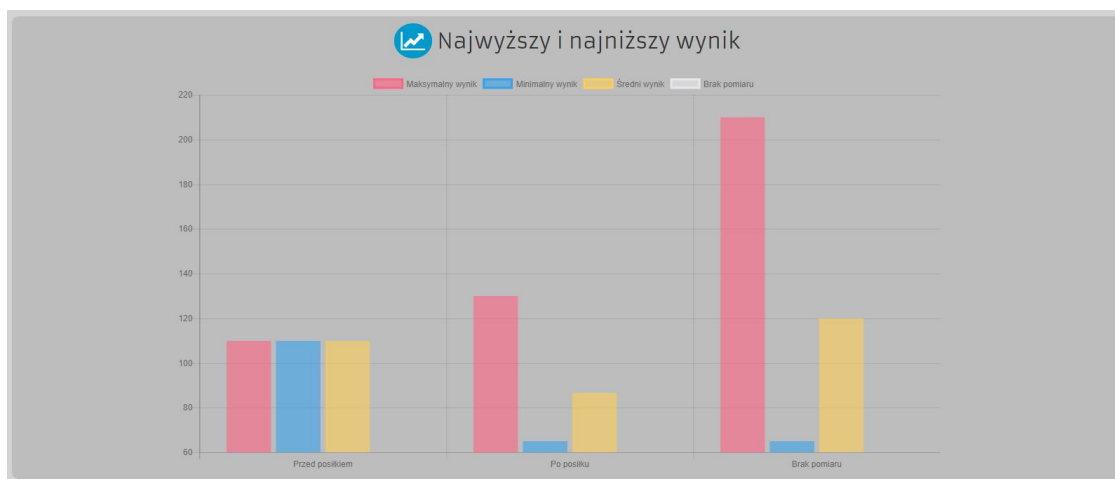
Dla lepszej przejrzystości, do zaznaczenia norm na wykresie zostały użyte charakterystyczne kolory. Przy przekroczeniu normy wykres przyjmuje kolor czerwony. Jeżeli pomiar glikemii mieścił się w normie, wykres zostaje zaznaczony kolorem zielonym. W przypadku wartości glikemii znajdującej się poniżej normy wykres przyjmuje kolor pomarańczowy. Dodatkowo, po najechaniu kursorem na daną część wykresu użytkownikowi wyświetla się dymek z podpowiedzią ile pomiarów znajduje się w danym zakresie norm.



Rysunek 6.4: Wykresy kołowe dostępne z poziomu zakładki *Twoja glikemia*

Kolejnym elementem dostępnym z poziomu zakładki *Twoja glikemia* są wykresy słupkowe. Zestawiają one maksymalny, minimalny oraz średni wynik dla każdego z rodzajów pomiarów glikemii. Po najechaniu na dany słupek wykresu użytkownik ma dostęp do informacji o wartości danego wyniku.

Ostatnim z elementów zakładki *Twoja glikemia* jest tabela zestawień. Podsumowuje ona (w zależności od pory dnia – rano, południe, wieczór, noc) następujące dane:



Rysunek 6.5: Wykresy słupkowe dostępne z poziomu zakładki *Twoja glikemia*

- Liczba wyników,
- Najwyższy wynik (mg/dL),
- Najniższy wynik (mg/dL),
- % wyników powyżej normy,
- % wyników poniżej normy,
- % wyników w normie.

Ostatnia kolumna tabeli przedstawia sumę wyników danego wiersza.

Czynnik główny	Rano	Południe	Wieczór	Noc	łącznie
Liczba wyników:	1	1	6	3	11
Najwyższy wynik (mgdL):	76	250	220	180	---
Najniższy wynik (mgdL):	76	250	65	45	---
% powyżej normy:	0.00%	100%	50.0%	33.3%	45.5%
% poniżej normy:	0.00%	0.00%	16.7%	33.3%	18.2%
% w normie:	100%	0.00%	33.3%	33.3%	36.4%

Rysunek 6.6: Tabela zestawień pomiarów glikemii dostępna z poziomu zakładki *Twoja glikemia*

Użytkownik ma możliwość wybrania przedziału czasu, na podstawie którego wyświetlane mają być dane. Dostępne są dwie opcje - *Ostatni miesiąc* oraz *Ostatni rok*. Wyboru dokonuje się poprzez kliknięcie jednego z dwóch dostępnych przycisków na szczycie podstrony.

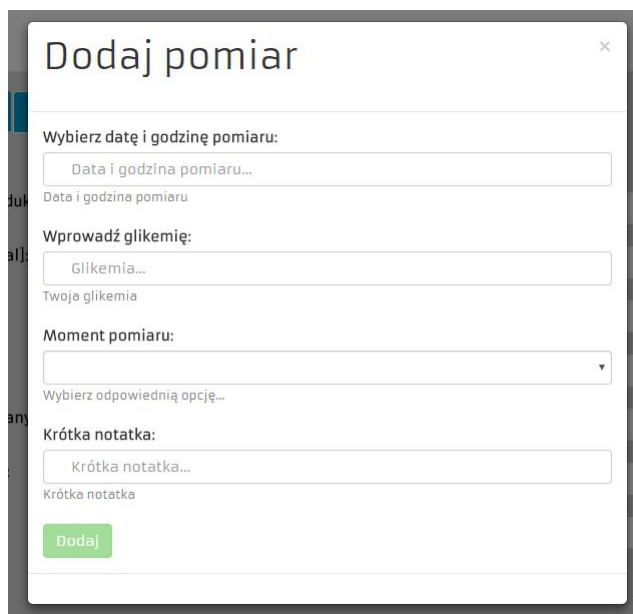


### 6.2.2 Dodaj pomiar

Zakładka *Dodaj pomiar* otwiera okno modalne dodawania nowego pomiaru (rys. 6.7). Z poziomu tego okna użytkownik wprowadza dane, które zestawiane są później na stronie komponentu *Twoja glikemia* w postaci wykresów i tabeli zestawień:

- Data i godzina pomiaru,
- Wartość glikemii,
- Moment pomiaru,
- Krótka notatka.

Po naciśnięciu przycisku *Dodaj* zostaje dodany do bazy danych pomiarów nowy pomiar.



Rysunek 6.7: Okno modalne dodawania nowego pomiaru

### 6.2.3 Kalkulatory

Zakładka *Kalkulatory* pozwala użytkownikowi na uzyskanie dostępu do kalkulatora *BMI (Body Mass Index)* oraz do kalkulatora wymienników węglowodanowych oraz białkowo-tłuszczowych, dostępnego wraz z tabelą makroskładników produktów spożywczych pochodzących z bazy danych produktów. Kalkulator *BMI (Body Mass Index)* pozwala obliczyć współczynnik masy do wzrostu. Składa się on z dwóch pól typu *Input*. W pierwszym z nich użytkownik wpisuje swoją wagę, a w następnym wzrost (rys. 6.8). Po kliknięciu przycisku *Oblicz* użytkownik otrzymuje informację o uzyskanym wyniku współczynnika *BMI* oraz odpowiedź do jakiej grupy osób (ze względu na wagę) się zalicza (rys. 6.9):

- osoby z niedowagą,
- osoby z wagą prawidłową,
- osoby z nadwagą,
- osoby z otyłością I stopnia,

- osoby z otyłością II stopnia,
- osoby z otyłością III stopnia.

Rysunek 6.8: Interfejs kalkulatora BMI

Rysunek 6.9: Interfejs kalkulatora BMI po uzyskaniu wyniku

Kolejną funkcjonalnością dostępną z poziomu zakładki *Kalkulatory* menu bocznego jest kalkulator wymienników węglowodanowych i wymienników białkowo-tłuszczowych na podstawie tabeli makroskładników produktów dostępnych z poziomu tabeli *products*, która może być stale rozszerzana przez zalogowanych użytkowników aplikacji (rys. 6.10). Wynik przedstawiany jest w postaci kolumn dołączonych do tabeli makroskładników danego produktu. Aby obliczyć wartość wymienników węglowodanowych i wymienników białkowo-tłuszczowych należy wybrać dany produkt poprzez wpisanie ciągu (bądź części ciągu) znaków w polu formularza o nazwie *Wybierz produkt*. Do tego pola przypięty jest inteligentny moduł wyszukiwania znajdujący produkty po słowach kluczowych. Po wpisaniu części wyrazu moduł podpowiada użytkownikowi najbardziej prawdopodobne wyniki w postaci listy rozwijanej. Dodatkowo użytkownik ma możliwość podania porcji produktu w polu *Wpisz wagę produktu [g]*. Wartości makroskładników i wyników obliczonych przez kalkulatory są wówczas dynamicznie zmieniane w zależności od wpisanej porcji produktu. Domyślna wartość pola to 100 [g].

Dodatkowo, jako ostatnia kolumna tabeli dodana została informacja, czy produkt może być szkodliwy dla zdrowia osoby chorej na cukrzycę, czy też nie. System rekomendacji oparty jest na ilości węglowodanów zawartych w produkcie. Jeżeli w 100 gramach produktu, 50 gram to węglowodany, wówczas produkt może okazać się szkodliwy w diecie cukrzyka (rys. 6.11).

**Kalkulator WW i WBT**

Wybierz produkt:

Wpisz wagę produktu [g]:

Tabela wartości:

Nazwa	Kcal	Białko [g]	Tłuszcz [g]	Węglowodany [g]	Błonnik [g]	Porcja [g]	WW	WBT	U
-------	------	------------	-------------	-----------------	-------------	------------	----	-----	---

Rysunek 6.10: Interfejs kalkulatora wymienników węglowodanowych i białkowo-tłuszczowych

**Kalkulator WW i WBT**

Wybierz produkt:

Wpisz wagę produktu [g]:

Tabela wartości:

Nazwa	Kcal	Białko [g]	Tłuszcz [g]	Węglowodany [g]	Błonnik [g]	Porcja [g]	WW	WBT	Uwagi
Bułka pszenna	277	8.10	1.50	57.7	1.80	100	5.77	0.444	Ten produkt może być szkodliwy dla Twojego zdrowia!

Rysunek 6.11: Interfejs kalkulatora wymienników węglowodanowych i białkowo-tłuszczowych po uzyskaniu wyniku

Użytkownik zarejestrowany ma również możliwość dodawania produktów do bazy danych produktów. Formularz dodawania nowego produktu dostępny jest również z poziomu zakładki *Twoja glikemia* – karta *Dodaj produkt* (rys. 6.12).

Po wypełnieniu wszystkich pól:

- Nazwa produktu,
- Kalorie [kcal],
- Białko [g],
- Tłuszcz [g],
- Węglowodany [g],
- Błonnik [g],
- Porcja [g]

produkt jest dynamicznie dodawany do bazy danych i bezpośrednio dostępny z pozycji modułu kalkulatora do obliczania wymienników węglowodanowych i białkowo-tłuszczowych.

Kalkulatory Dodaj produkt

## Dodaj produkt

Nazwa produktu:

Kalorie [kcal]:

Białko [g]:

Tłuszcz [g]:

Węglowodany [g]:

Błonnik [g]:

Porcja [g]:

Zapisz

Rysunek 6.12: Formularz dodawania nowego produktu

### 6.2.4 Twój profil

Z poziomu zakładki *Twój profil* użytkownik ma możliwość dodania informacji zarówno o profilu (rys. 6.13) jak i danych medycznych (rys. 6.14). Czynność ta odbywa się przy pomocy dwóch formularzy z odpowiednimi polami:

- Dla karty *Twój profil*
  - imię,
  - nazwisko,
  - PESEL,
  - opiekun - lista rozwijana,
  - ulica,
  - nr domu,
  - miasto,
  - kod pocztowy,
  - numer telefonu,
  - aktualny glukometr - lista rozwijana,
  - poprzedni glukometr - lista rozwijana,
  - numer seryjny glukometru.
- Dla karty *Dane medyczne*
  - typ cukrzycy - lista rozwijana,
  - rok zachorowania - lista rozwijana,
  - rodzaj insuliny - lista rozwijana,
  - przyjmowany lek - lista rozwijana,
  - inne leki i insuliny,
  - BMI,

- wzrost,
- waga,
- hBA1c.

Przy pierwszym wpisaniu danych do wszystkich pól formularza i zatwierdzeniu ich przyciskiem *Zapisz*, zostanie wykonana operacja zapisania danych do obydwu tabel bazy danych. Przy każdorazowym otwarciu zakładki *Twój profil* dane te będą czytane z bazy danych do pól formularza. W przypadku chęci zaktualizowania informacji użytkownik nadpisuje pola formularza nowymi danymi i wykonywana jest operacja *UPDATE* zarówno w przypadku jednej, jak i drugiej tabeli.

Twój profil		Dane medyczne	
Imię:	<input type="text" value="Patryk"/>		
	Wpisz swoje imię...		
Nazwisko:	<input type="text" value="Dzwoniarski"/>		
	Wpisz swoje nazwisko...		
PESEL:	<input type="text" value="94061401032"/>		
	Wpisz swój numer PESEL...		
Opiekun:	<input type="text" value="Tak"/>		
	Wybierz odpowiednią opcję...		
Ulica:	<input type="text" value="Poznańska"/>		
	Wpisz swoją ulicę...		
Nr domu:	<input type="text" value="12"/>		
	Wpisz swój numer domu...		
Miasto:	<input type="text" value="Słupca"/>		
	Wpisz nazwę swojego miasta		
Kod pocztowy:	<input type="text" value="62-400"/>		
	Wpisz kod pocztowy Twojego miasta...		
Numer telefonu:	<input type="text" value="608456777"/>		
	Wpisz swój numer telefonu...		
Aktualny glukometr:	<input type="text" value="Accu-Check Active"/>		
	Wybierz odpowiednią opcję...		
Poprzedni glukometr:	<input type="text" value="Accu-Check Performa"/>		
	Wybierz odpowiednią opcję...		
Numer seryjny glukometru:	<input type="text" value="AHBHC-234AC"/>		
	Wpisz numer seryjny Twojego glukometru...		
<input type="button" value="Zapisz"/>			

Rysunek 6.13: Formularz dodawania i aktualizacji informacji o profilu

Twój profil   Dane medyczne

Typ cukrzycy: Typ 2

Rok zachorowania: 2009

Rodzaj insuliny: NovoMix 30 Penfil

Przyjmowany lek: Adeksa, tabl., 100 mg

Inne leki i insuliny: Ibuprofen

BMI: 28.00

Wzrost: 187

Waga: 79.00

HbA1c: 114.00

Zapisz

Rysunek 6.14: Formularz dodawania i aktualizacji danych medycznych użytkownika profilu

### 6.2.5 Ustawienia profilu

W zakładce *Ustawienia profilu* użytkownik ma możliwość dodania, bądź zaktualizowania bieżących zakresów glikemii (rys. 6.15). Komponent składa się z trzech pól formularza:

- Hipoglikemia,
- Hiperglikemia,
- Hiperglikemia po posiłku.

Ustawienia profilu - zakresy glikemii

Hipoglikemia\*: 101

Hiperglikemia\*: 202

Hiperglikemia po posiłku\*: 303

Zapisz

Rysunek 6.15: Formularz dodawania i aktualizacji danych dotyczących zakresów glikemii użytkownika

Każde pole może być w dowolnym czasie aktualizowane (operacja *UPDATE*) w bazie danych. Ponadto dane wczytywane są bezpośrednio do pól formularza przy użyciu metody *GET*.

## Rozdział 7

# Panel użytkownika niezarejestrowanego oraz niezalogowanego

### 7.1 Okno rejestracji użytkownika

W poniższej sekcji ukazany zostanie interfejs programu umożliwiający użytkownikowi niezarejestrowanemu w systemie dokonania rejestracji. Rejestracja przebiegać może w dwóch wariantach:

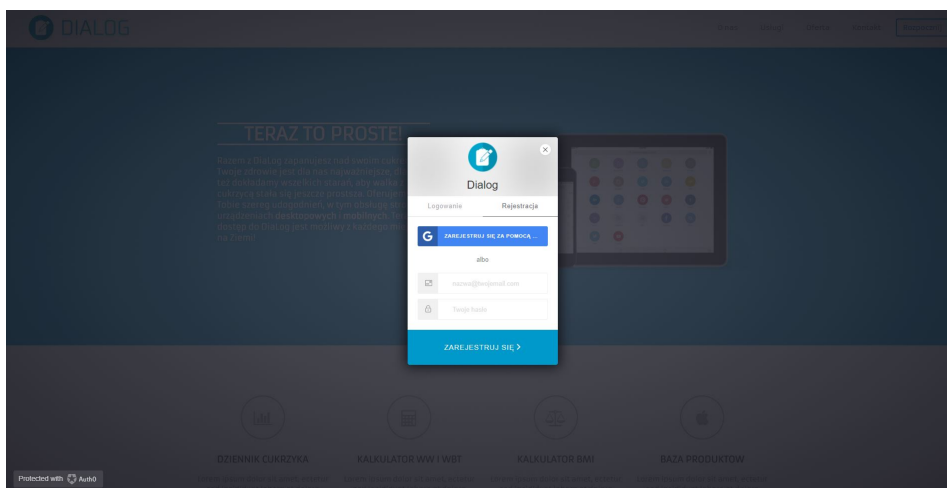
- Rejestracja przy użyciu konta *Google*,
- Rejestracja tradycyjna – podanie adresu e-mail oraz hasła.

#### 7.1.1 Rejestracja przy użyciu konta *Google*

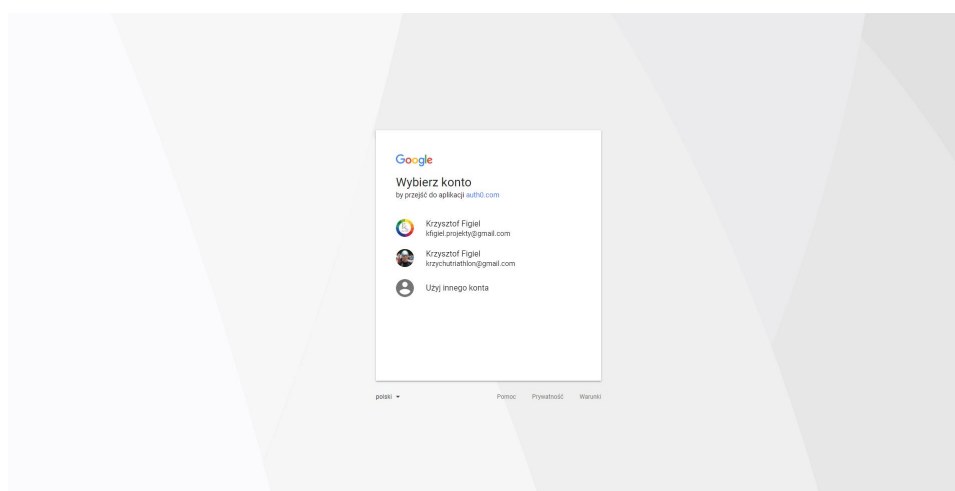
Opcja rejestracji i logowania za pomocą konta *Google* znacznie ułatwia i przyspiesza proces dodawania nowego użytkownika. Pozwala ona zwiększyć odsetek nowych użytkowników systemu. Kolejnym powodem, który przemawia za tym, aby stosować logowanie przy użyciu danych pochodzących z portali społecznościowych jest fakt mówiący o tym, że adres e-mail, którym posługuje się dana osoba w momencie takiej rejestracji został już wcześniej zweryfikowany przez dostawcę sieci społecznościowej. Oznacza to, że w momencie logowania przy użyciu konta społecznościowego otrzymujemy rzetelne informacje, a nie fałszywe adresy, które użytkownicy wykorzystują zazwyczaj do zarejestrowania się na stronach internetowych.

Cały proces rejestracji i późniejszego logowania się do aplikacji przy użyciu konta *Google* przebiega w następujący sposób:

- Użytkownik wybiera opcję rejestracji w systemie i klika przycisk *Zaloguj się za pomocą konta Google* (rys. 7.1) i wpisuje dane logowania do konta *Google*, bądź w przypadku kiedy jest zalogowany globalnie wybiera odpowiednie konto (rys. 7.2),
- Żądanie rejestracji/logowania wysłane jest do dostawcy sieci społecznościowej *Google*,
- W momencie gdy dostawca sieci społecznościowej *Google* potwierdzi tożsamość użytkownika, bieżący użytkownik uzyskuje dostęp do aplikacji.



Rysunek 7.1: Okno rejestracji użytkownika do systemu

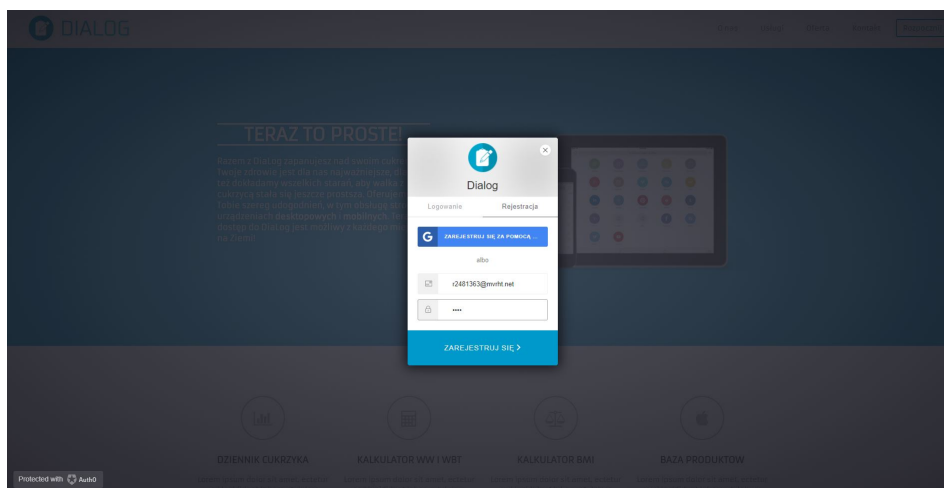


Rysunek 7.2: Okno wyboru konta Google

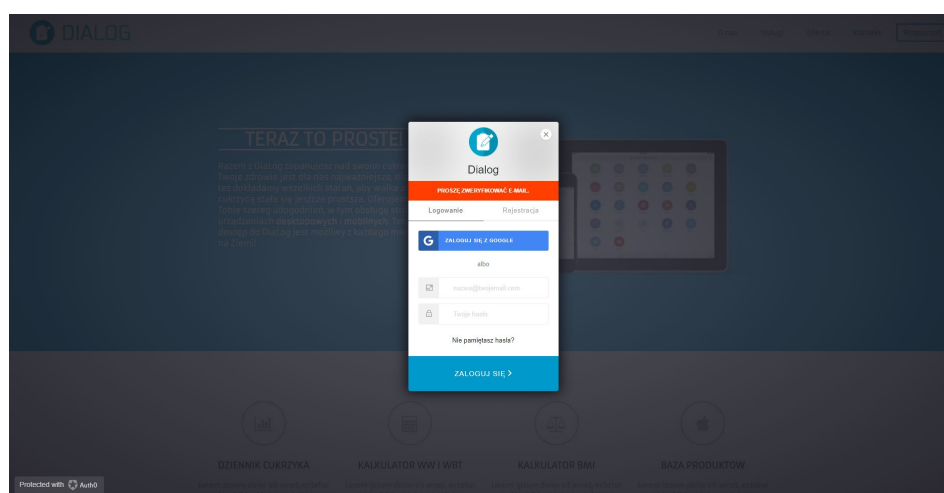
### 7.1.2 Rejestracja tradycyjna – podanie adresu e-mail oraz hasła

W przypadku rejestracji tradycyjnej dane podane przez użytkownika (adres e-mail oraz hasło) zostają zapisane w zewnętrznej bazie danych dostawcy usług autoryzacji – *Auth0*. Po kliknięciu przycisku *Zarejestruj się* (rys. 7.3) użytkownik zostaje powiadomiony o konieczności potwierdzenia autentyczności konta poprzez kliknięcie w link aktywacyjny wysłany automatycznie w wiadomości e-mail przez serwis *Auth0* (rys. 7.4). Po kliknięciu w link aktywacyjny możliwe jest już zalogowanie się.





Rysunek 7.3: Okno rejestracji tradycyjnej



Rysunek 7.4: Okno z prośbą o potwierdzenie adresu e-mail

Po poprawnej rejestracji i późniejszym zalogowaniu się użytkownik ma dostęp do panelu użytkownika zarejestrowanego. Z poziomu panelu użytkownika niezalogowanego możliwe jest również uzyskanie informacji na temat funkcjonalności strony oraz wysłanie maila z zapytaniem do administracji za pomocą wbudowanego formularza z podpiętym modulem do wysyłania wiadomości e-mail. Użytkownik może w każdej chwili wypełnić wymagane pola:

- Imię,
- Nazwisko,
- Temat,
- Wiadomość

i wysłać wiadomość do administracji z dowolnym zapytaniem, klikając przycisk *Wyślij* (rys. 7.5).

The screenshot displays the DIALOG website interface. At the top, there is a navigation bar with the DIALOG logo and links for 'O nas', 'Usługi', 'Oferta', 'Kontakt', 'Twój Dialog', and a 'Wyloguj' button. Below the navigation bar, the main content area is divided into two sections. On the left, there is a map showing a street grid with labels like 'Szujskiego 66' and 'Szujskiego 66, 33-100 Tarnobrzeg'. A red pin is placed on the map. On the right, there is a contact form with the following fields: 'Jan', 'Nowak', 'Jak dodać pomiary?', and a large text area containing the text: 'Witam, jak dodaje się pomiary w Pańskiej aplikacji? Pozdrawiam, Jan Nowak'. A green 'Wyślij' button is located at the bottom of the form.

Rysunek 7.5: Formularz do wysyłania maili do administracji

## Rozdział 8

# Bezpieczeństwo aplikacji

Do przechowywania wrażliwych danych użytkownika, takich jak hasło wykorzystywane są mechanizmy oferowane przez usługodawcę *Auth0*. Dane przechowywane w bazie tego serwisu nie mają postaci zwykłego tekstu, a przechowywane za pomocą funkcji skrótu *BCrypt*.

*BCrypt* jest funkcją skrótu kryptograficznego, która została stworzona w celu przechowywania haseł statycznych, czyli haseł znanych wyłącznie osobie, która chce się uwierzytelnić, a nie dowolnych danych binarnych. *BCrypt* wymaga stosowania soli, co wyróżnia ją od innych funkcji skrótu. Sól w algorytmie *BCrypt* jest złożona z następujących elementów:

- *version* oznaczające wersję algorytmu *BCrypt*,
- *rounds* jest to liczba z przedziału 04-99, która określa tzw. *work factor* algorytmu, domyślna wartość tego pola to \$12,
- *saltaddon* – są to losowe 22 znaki, które mają za zadanie powiększać sól weryfikacja tego ciągu przebiega przy użyciu wyrażenia regularnego `[./A-Za-z0-9]` – znaki te mogą być wylosowane przez użytkownika, bądź przez specjalnie zaprojektowany do tego celu algorytm.

W *Auth0* zarówno dane *REST-owe*, jak i przekazywane w ruchu sieciowym są szyfrowane. Cała komunikacja sieciowa wykorzystuje protokół TLS (*Transport Layer Security*) w wersji 1.2 (będący rozwinięciem protokołu SSL), z conajmniej 128-bitowym szyfrowaniem AES (*Advanced Encryption Standard*). Do wymiany kluczy wykorzystywany jest mechanizm *ECDHE\_RSA* oparty na protokole *Diffiego-Hellmana* z podpisem generowanym przy użyciu kryptograficznego algorytmu asymetrycznego RSA (*Rivest-Shamir-Adleman*).

Usługi świadczone przez serwis *Auth0* zaprojektowane są z myślą o wysokiej dostępności i odporności. Aplikacje korzystające z *Auth0* są częściowo zabezpieczone przed atakami typu *Odmowa usługi* czy *Uwierzytelnianie*. Mają one wbudowane funkcje ograniczania szybkości i automatycznego blokowania. Ponadto konta użytkowników zabezpieczone są za pomocą domyślnie wbudowanego modułu weryfikacji autentyczności użytkownika przy użyciu adresu e-mail. Każdy użytkownik systemu otrzymuje unikalny JWT (*JSON Web Token*), który pozwala na rozróżnianie i sprawdzanie autentyczności użytkowników aplikacji. Aplikacja została zabezpieczona przed skopiowaniem linku do panelu użytkownika zalogowanego i wklejeniu go do innego okna przeglądarki w celu uzyskania dostępu. W momencie takiej próby użytkownik zostaje poinformowany, że nie ma uprawnień do przeglądania treści danej strony (rys. 8.1).



Rysunek 8.1: Błąd przy próbie uzyskania dostępu do panelu użytkownika zalogowanego poprzez link URL

Jak widać u góry (w menu górnym) nie ma dostępu do zakładki *Twój dialog*, co oznacza, że użytkownik jest niezalogowany i próbuje uzyskać dostęp do linku URL udostępnianego wyłącznie użytkownikowi zalogowanemu.

Podobna sytuacja dotyczy się panelu administratora. Użytkownik nie posiadający w serwisie *Auth0* roli *Admin* nie posiada dostępu do funkcjonalności dla niej przeznaczonych. To oznacza, że nie da się uzyskać bezpośredniego dostępu do podstrony panelu administratora poprzez wklejenie linku URL do okna wyszukiwarki. Użytkownik przy takiej próbie zostaje natychmiastowo informowany właściwym komunikatem (rys. 8.2).



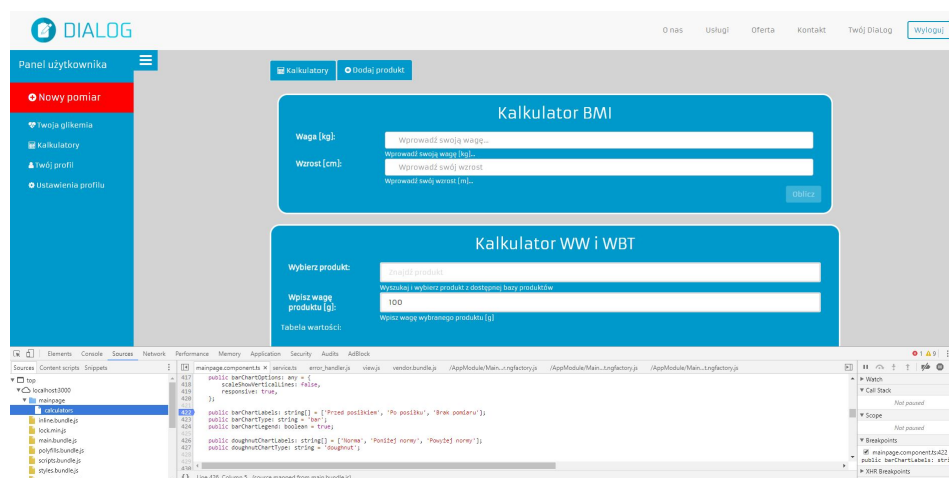
Rysunek 8.2: Błąd przy próbie uzyskania dostępu do panelu administratora poprzez link URL

## Rozdział 9

# Testowanie aplikacji

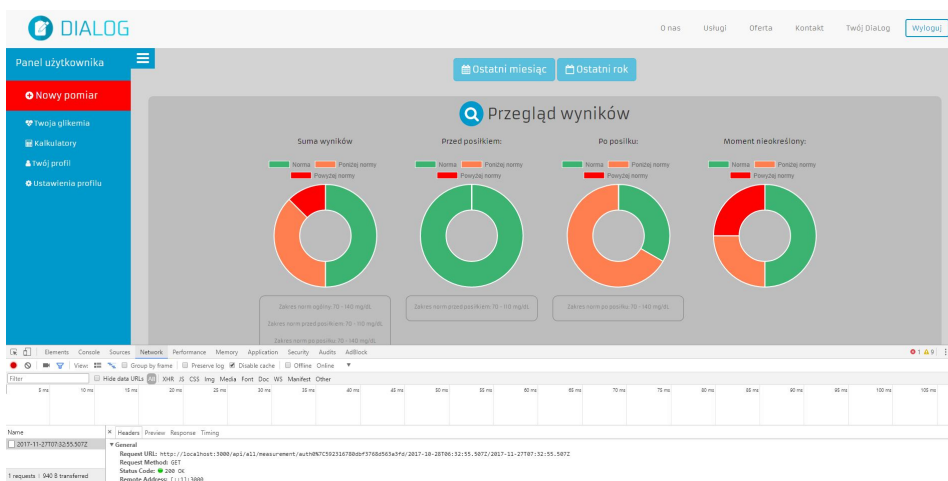
### 9.1 Debugowanie aplikacji i dostęp do informacji o jej działaniu

Do wykonywania debugowania aplikacji w celu znalezienia błędów użyto wtyczki do programu *Visual Studio Code* oraz wbudowanego w przeglądarkę internetową *Google Chrome*, debuggera dostępnego z poziomu panelu narzędzi deweloperskich (zakładka *Sources*) widoczna na rysunku 9.1. W tym miejscu możliwe jest podejrzenie i debugowanie skryptów aplikacji. Programista ma również możliwość śledzenia wybranych wyrażeń (pole *Watch expressions*), przeglądania stosu aplikacji (pole *Call Stack*), ustawiania tzw. *breakpointów* w momencie debugowania, przechodzenia kodu krok po kroku i wiele, wiele innych.



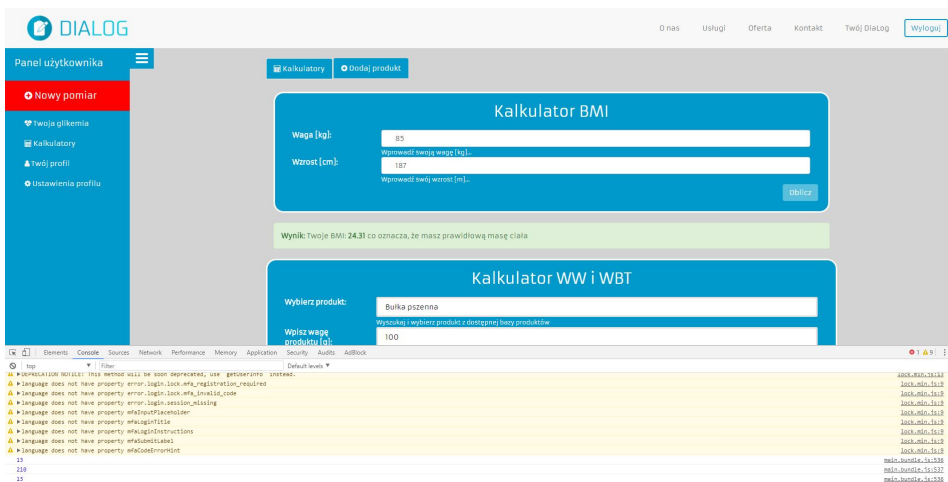
Rysunek 9.1: Opcje debugowania i podglądu kodu dostępne z poziomu zakładki *Sources* narzędzi deweloperskich przeglądarki *Google Chrome*

Do podglądu poprawności wysyłanych żądań za pomocą metod HTTP użyto narzędzi dostępnych z poziomu zakładki *Network*. Oferuje ona podgląd adresów URL (*Uniform Resource Locator*) generowanych do pobrania danych w formacie JSON oraz treści tablicy obiektu zawierającego te dane (rys. 9.2).



Rysunek 9.2: Opcje podglądu przesyłanych żądań dostępne z poziomu zakładki *Network* narzędzi deweloperskich przeglądarki *Google Chrome*

W celu identyfikacji rodzaju błędu posłużono się oknem konsoli dostępnym z poziomu zakładki *Console*. Oferuje ona dokładny opis zaistniałego błędu wraz ze ścieżką lokalizacji do pliku z błędnym fragmentem kodu, a nawet numeru liniiki, w której dany błąd wystąpił. W przypadku użycia metod HTTP (*Hypertext Transfer Protocol*) wyświetlany jest odpowiedni numer błędu zgodnie ze specyfikacją kodów błędów HTTP. W oknie konsoli możliwy jest również podgląd wartości przechowywanych przez zmienne w projekcie oraz dostęp do treści ostrzeżeń w trakcie działania aplikacji (rys. 9.3).



Rysunek 9.3: Opcje podglądu błędów, ostrzeżeń i wartości zmiennych dostępne z poziomu zakładki *Console* narzędzi deweloperskich przeglądarki *Google Chrome*

## 9.2 Wydajność aplikacji

Aplikacja skonstruowana jest pod kątem UX (*User Experience*), a kod pisany był w oparciu o podstawowe zasady budowania struktury projektu tworzonego z wykorzystaniem frameworku *Angular 2*. [Ans] Dołożono wszelkich starań, aby ilość danych umiesz-

czanych w ramach jednego widoku nie była zbyt duża, ponieważ w przypadku *Angulara 2* znacząco obniża to wydajność aplikacji.

Kolejnym ważnym czynnikiem znacząco wpływającym na wydajność jest to, że dane w obrębie widoku są odświeżane w sposób dynamiczny – okno przeglądarki nie wymaga odświeżenia w celu zmiany swojej zawartości. Ponadto, formularze dostępne na stronie oparte są o technologię *Reactive Forms*. Dzięki niej nie ma konieczności tworzenia dużej liczby zmiennych przechowujących dane pól formularza i dwukierunkowej zmiany ich wartości z poziomu komponentu oraz widoku. W zamian za to użytkownik ma możliwość stworzenia drzewa *Angularowych* obiektów kontrolujących pola formularza w klasie komponentu i połączenia ich z elementami sterującymi formularza w szablonie danego komponentu. Sama manipulacja wartościami obiektów sterujących odbywa się tylko i wyłącznie z poziomu klasy komponentu, który obserwuje zmiany i reaguje na każdą z nich w czasie rzeczywistym. Ważną zaletą stosowania tego typu formularzy jest to, że aktualizacja wartości pól formularza odbywa się w sposób synchroniczny i jest stale pod kontrolą programisty. [Ang]





## Rozdział 10

# Zakończenie

Ustosunkowując się do założeń podanych we wstępie pracy, dołożono wszelkich starań, aby oferowane przez aplikację funkcjonalności działały zgodnie z ich przeznaczeniem oraz zaprezentowane były za pomocą prostego i przejrzystego interfejsu. Aplikacja ma być jedynie jednym z czynników przyczyniających się do postępów w leczeniu cukrzycy.

Poprzez oferowanie czytelnych wykresów podsumowujących wszystkie, dotychczasowe pomiary wprowadzone przez użytkownika oraz tabeli zestawień grupującej wartości pomiarów osoba korzystająca z aplikacji wyciągać ma poglądowe wnioski na temat tego jak przebiega jej choroba od momentu założenia konta w systemie. Ponadto, dzięki dostępnym kalkulatorom BMI (*Body Mass Index*), wymienników węglowodanowych czy wymienników białkowo-tłuszczowych opartych na stale rozwijanej przez użytkowników bazie produktów i danych wprowadzanych przez korzystającą z nich osobę możliwe jest określenie jaki stopień współczynnika masy ciała posiada dany użytkownik, czy też jakie produkty mogą być spożyte w danej chwili tak, aby nie zakłócić porządku dziennej diety i nie pogłębić stopnia zaawansowania cukrzycy.

Jeżeli chodzi o dalsze etapy rozwoju aplikacji – autor chciałby rozszerzyć jej funkcjonalność o moduł oparty na zdobywaniu punktów przez użytkowników w zamian za postępy w leczeniu cukrzycy. Funkcjonalność ta mogłaby mieć znaczący wpływ na motywację i sukcesywne korzystanie z aplikacji, a co za tym idzie systematyczne prowadzenie dziennika pomiarów. Dodatkowo, jest w planach dodanie modułu skanowania kodów kreskowych produktów przez użytkowników za pomocą kamerki internetowej, bądź kamery telefonu i automatycznego dodawania ich do bazy danych systemu. Ułatwiłoby to znacząco proces wprowadzania nowego produktu.



# Literatura

- [Ang] Angular - Reactive Forms guide. [on-line 11.11.2017]  
<https://angular.io/guide/reactive-forms>.
- [Ans] Angular - Setup for local development guide. [on-line 13.08.2017]  
<https://angular.io/guide/setup>.
- [Aut] Auth0 documentation. [on-line 01.10.2017]  
<https://auth0.com/docs/quickstart/spa/angular2>.
- [Car17] Patrick Carey. *New Perspectives on HTML5 and CSS3, 7th Edition*. Cengage Learning., Boston MA 02210, 20 Channel Street B3 2PB, USA, 2017.
- [Dee16] Pablo Deeelman. *Learning Angular 2*. Packt Publishing Ltd., Birmingham, 35 Livery Street B3 2PB, UK, 2016.
- [Duc15] Jon Duckett. *JavaScript i jQuery — Interaktywne strony WWW dla każdego*. Wydawnictwo HELION, Gliwice, ul. Kościuszki 1c, 44-100 Gliwice, PL, 2015.
- [Fra15] Ben Frain. *Responsive Web Design with HTML5 and CSS3 — Second Edition*. Packt Publishing Ltd., Birmingham, 35 Livery Street B3 2PB, UK, 2015.
- [Les] LESS documentation. [on-line 13.10.2017] <http://lesscss.org/>.
- [Mys] MySQL Workbench documentation. [on-line 20.09.2017] <https://dev.mysql.com/doc/>.
- [Nan14] Christopher Nance. *TypeScript Essentials*. Packt Publishing Ltd., Birmingham, 35 Livery Street B3 2PB, UK, 2014.
- [Nod] Node.js guides. [on-line 13.11.2017] <https://nodejs.org/en/docs/guides/>.
- [Rah14] Syed Fazle Rahman. *Jump Start Bootstrap*. Site Point Pty. Ltd., Victoria, 48 Cambridge Street Collingwood, AUS, 2014.
- [Seq] Sequelize manual. [on-line 08.11.2017] <http://docs.sequelizejs.com/>.
- [Vsc] Visual Studio Code documentation. [on-line 08.09.2017]  
<https://code.visualstudio.com/docs>.



## Dodatek A

# Załączniki

Płyta CD z następującą zawartością:

- tekst pracy w formacie PDF,
- projekt z plikami źródłowymi.