

**UNIwersytet Warmiński-Mazurski**  
**W OLSZTYNIE**  
**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**Kierunek: Informatyka**

**Przemysław Dzieńisiewicz**

**Aplikacja do robienia sprawnych zakupów**

Praca inżynierska wykonana  
w Katedrze Metod  
Matematycznych Informatyki  
pod kierunkiem  
*dr Pawła Drozdy*

Olsztyn 2021

**UNIVERSITY OF WARMIA AND MAZURY  
IN OLSZTYN  
FACULTY OF MATHEMATICS AND COMPUTER  
SCIENCE**

**Field of study: Computer Science**

**Przemysław Dzienisiewicz**

**Efficient shopping application**

Engineering Thesis written in  
Department of Mathematical  
Methods in Computer Science  
under supervision of  
*dr Paweł Drozda*

Olsztyn 2021

## Spis treści

1. Wstęp .....	4
2. Opis technologii .....	6
3. Struktura pracy .....	10
3.1. Schemat bazy danych .....	10
3.2. Opis bazy danych.....	11
3.3. Diagram przypadków użycia .....	12
3.3.1. Uogólniony diagram przypadków użycia .....	12
3.3.2. Rozwinięcie przypadku użycia „Zarządzaj aktualną listą zakupów”	13
3.3.3. Rozwinięcie przypadku użycia „Zarządzaj bazą produktów” .....	17
3.3.4. Rozwinięcie przypadku użycia „Wyszukaj restauracje” .....	22
3.3.5. Rozwinięcie przypadku użycia „Zarządzaj kontem” .....	26
4. Dokumentacja użytkownika .....	30
Ekran rejestracji do aplikacji.....	30
Ekran logowania do aplikacji .....	32
Ekran aktualnej listy zakupów – pusta lista .....	35
Ekran bazy produktów użytkownika – pusta lista .....	38
Ekran bazy produktów użytkownika – uzupełniona lista.....	42
Ekran aktualnej listy zakupów – uzupełniona lista.....	44
Ekran wyszukiwania restauracji .....	46
Ekran wyszukiwania restauracji – ekran dodatkowych informacji o restauracji.....	50
Ekran zarządzania kontem.....	51
5. Podsumowanie .....	52
Streszczenie.....	53
Abstract.....	53
Bibliografia .....	54
Spis Tabel .....	56
Spis zdjęć.....	57
Spis diagramów.....	58

# 1. Wstęp

Tematem pracy inżynierskiej jest aplikacja mobilna o nazwie „Shopping App”, która powstała w celu usprawnienia zakupów. Użytkownik aplikacji jest w stanie znacząco przyspieszyć swoje zakupy dzięki zaplanowaniu kupna potrzebnych mu produktów – poprzez wprowadzenie ich na listę w aplikacji tuż przed wyruszeniem do sklepu. Aby zakupy przebiegały jeszcze sprawniej, w trakcie może odznaczać zakupione już artykuły. Posiadacz aplikacji ma bazę swoich produktów zawsze pod ręką - i jeśli chce – może swobodnie wracać do wcześniej dodanych artykułów, nawet gdy na aktualnej liście nie będzie ich potrzebował.

Aplikacja została stworzona przez Przemysława Dzienisiewicza, który odpowiedzialny był za całokształt. Motywacją do jej powstania była istota poszerzenia wiedzy z zakresu JavaScript’owej biblioteki ‘React Native’ służącej do tworzenia interfejsów użytkownika, która współdzieli zawartość samego ‘React.js’, umożliwiając programistom korzystanie z frameworka wraz z natywną platformą (tzn. konkretnym systemem operacyjnym tj. android, iOS). Oprócz samej nauki, wyszła także potrzeba posiadania aplikacji na własny użytek, która pozwoli na sprawniejszą realizację zakupów - przede wszystkim uporządkowania samej listy produktów oraz skrócenia czasu ich trwania, szczególnie w aktualnym czasie pandemii, przez co użytkownik może zminimalizować potencjalny kontakt z ludźmi przebywającymi w danym sklepie.

Tworzenie aplikacji nie byłoby możliwe bez możliwości podglądu jej aktualnego stanu oraz fizycznego sprawdzenia działania. Za tę kwestię odpowiadał prywatny smartphone (Samsung S8) oraz środowisko ‘Expo’ uruchamiane na komputerze osobistym, które działało jako emulator urządzenia mobilnego.

W aplikacji do stworzenia różnych ekranów użytkownika zostały użyte komponenty wielokrotnego użytku, które zapewnia zestaw z samej paczki biblioteki 'React Native' oraz 'React Native Elements'. Zbudowane w całości w JavaScript są spójnym zestawem dla wielu platform tj. android, iOS oraz przeglądarek. Zawierają predefiniowane style wielu komponentów (np. przyciski), toteż zdecydowanie ułatwiają one pracę nad tworzeniem aplikacji.

Na całym etapie tworzenia aplikacji używany był system kontroli wersji GIT na platformie 'GitHub', dzięki któremu możliwe było ciągłe śledzenie zmian w kodzie źródłowym, co usprawniło pracę z uwagi na potrzeby zmian, bądź powrotu do poprzednich wersji aplikacji, odtworzenia sprawnego stanu projektu lub porównania wprowadzonych zmian.

Pomocne okazało się również 'Trello', czyli aplikacja internetowa do tworzenia list w stylu Kanban (tzw. metoda sterowania wytwarzaniem oprogramowania), która pozwalała na zwinne zarządzanie ustalonymi zadaniami oraz poszczególnymi etapami powstawania aplikacji od samego początku.

## 2. Opis technologii

**React Native** – główny filar na którym opiera się aplikacja. Jest to technologia stworzona przez firmę „Facebook” i służy do tworzenia wieloplatformowych aplikacji mobilnych (wspólny kod dla wszystkich platform - android oraz iOS), co zarazem jest jedną z jej największych zalet. Drugą jest natomiast to, iż całą aplikację piszemy w języku JavaScript – od logiki, aż po widoki.

**Expo** – środowisko ułatwiające pracę z React Native. W tym projekcie używane było jak emulator urządzenia mobilnego, pozwalającego na pokazywanie efektów pracy w czasie rzeczywistym podczas programowania. W jej szereg wchodzi kilka ważnych rzeczy, tj.:

- Aplikacja desktopowa, która umożliwia budowanie aplikacji, debugowanie
- Aplikacja mobilna, którą można zainstalować na dowolnym urządzeniu mobilnym z androidem lub iOS i odpalić przez nią stworzoną aplikację.
- Biblioteki expo, które zapewniają wiele pomocniczych rozwiązań ułatwiających tworzenie aplikacji (np. obsługa powiadomień)

**JSX** – (JavaScript XML) to format zapisu kodu HTML oraz XML wewnątrz języka JavaScript. Pierwotnie został zaproponowany przez Facebook’a i użyty w ich bibliotece ‘React.js’. Aktualnie korzystają z niego również inne biblioteki. Zalicza się jako ‘lukier składniowy’ (ang. ‘syntactic sugar’), czyli cecha składni języka, którą można wyeliminować przez proste przekształcenia składniowe, istniejąca jedynie dla wygody programisty podczas pracy.

**CSS3** – (ang. Cascading Style Sheets) tłumaczone jako kaskadowe arkusze stylów – język używany do opisu prezentacji dokumentu napisanego w języku znaczników, takim jak HTML (nadawanie stylów). Służy do kontrolowania wyglądu poszczególnych elementów na stronie lub aplikacji (tj. położenie obrazów, tekstu). Jedną z trzech podstawowych technologii WWW – obok JavaScript oraz HTML.

**HTML5** – język znaczników, którego używa się do strukturyzowania i nadawania znaczenia naszej zawartości internetowej, na przykład definiowania akapitów, nagłówek oraz tabel danych.

**JavaScript** – (JS) język programowania zgodny ze specyfikacją ECMAScript, który umożliwia implementację złożonych funkcji na stronach internetowych lub w aplikacjach – odpowiada min.in. za dynamiczne aktualizowanie treści, sterowanie multimediami oraz interaktywne funkcje. Jest językiem wysokopoziomowym, często kompilowanym na czas oraz wieloparadygmatycznym.

**MongoDB** – jest otwartym, nierelacyjnym systemem zarządzania bazą danych. Jego głównymi zaletami są duża skalowalność, wydajność oraz brak ściśle zdefiniowanej struktury obsługiwanych baz danych. Wykorzystuje dokumenty podobne do JSON z opcjonalnymi schematami.

**Visual Studio Code** – darmowy, usprawniony edytor kodu z obsługą operacji programistycznych tj. debugowanie, uruchamianie zadań i kontrola wersji – stworzony przez firmę Microsoft dla Windows, Linux oraz macOS. Jest w pełni konfigurowalny pod potrzeby użytkownika.

**Android Studio** – program zapewniający ujednolicone środowisko, w którym można tworzyć aplikacje na telefony i tablety z systemem android. Pozwala na uruchomienie urządzenia wirtualnego systemu android (emulator) do testowania i debugowania aplikacji.

**Git Bash** - aplikacja terminalowa dla środowisk Microsoft Windows, która zapewnia warstwę emulacji dla obsługi poleceń Git. Pozwalała na łączenie się z systemem operacyjnym za pomocą pisemnych poleceń.

**GitHub** – platforma hostingu kodu do tworzenia oprogramowania oraz kontroli wersji za pomocą Git. Oferuje funkcje rozproszonej kontroli wersji i zarządzania kodem źródłowym Git, a także własne funkcje – tj. śledzenie błędów, zarządzanie zadaniami, ciągłą integrację. Github ułatwił szczególnie pracę nad aplikacją w przypadku programowania na dwóch różnych urządzeniach – wymiana aktualnym kodem źródłowym była bardzo sprawna.

**Yelp** – serwis społecznościowy umożliwiający użytkownikom publikowanie recenzji i ocenianie firm(tj. restauracje, butiki, salony). Przedsiębiorstwa mogą aktualizować swoje dane kontaktowe, ofertę oraz zdjęcia usług i towarów. W Polsce głównymi miejscami ocenianymi w serwisie są Warszawa i Kraków. Umożliwił zaimplementowanie wyszukiwania oraz uzyskiwania informacji o restauracji w aplikacji.

**Mongoose** – narzędzie do modelowania obiektów MongoDB zaprojektowane do pracy w środowisku asynchronicznym. Zawiera proste, oparte na schemacie rozwiązanie do modelowania danych aplikacji – zawiera wbudowane rzutowanie typów, walidację, budowanie zapytań czy podpięcia logiki biznesowej. Ułatwiło pracę z bazą danych.



**Nodemon** – narzędzie, które pomaga w tworzeniu aplikacji opartych na node.js poprzez automatyczne ponowne uruchamianie aplikacji 'node' po wykryciu zmian plików w katalogu. Umożliwia płynną pracę podczas działania z emulatorem środowiska aplikacji mobilnej.

**Ngrok** – prosta w obsłudze, wieloplatformowa aplikacja, która umożliwia udostępnienie lokalnego serwera programistycznego w Internecie. Oprogramowanie sprawia, że lokalnie hostowany serwer WWW jest hostowany w poddomenie ngrok.com, co oznacza, że nie jest potrzebny publiczny adres IP ani nazwa domeny na komputerze lokalnym. Pozwolił na darmową pracę z aplikacją mobilną i połączenie do bazy danych w MongoDB.

**Trello** – internetowa aplikacja do tworzenia list w stylu Kanban, pozwalająca na zwinne zarządzanie zadaniami.

**Windows PowerShell** – wieloplatformowy, rozbudowany interpreter poleceń do automatyzacji zadań i zarządzania konfiguracją, składający się z powłoki wiersza poleceń i języka skryptowego.

### 3. Struktura pracy

#### 3.1. Schemat bazy danych

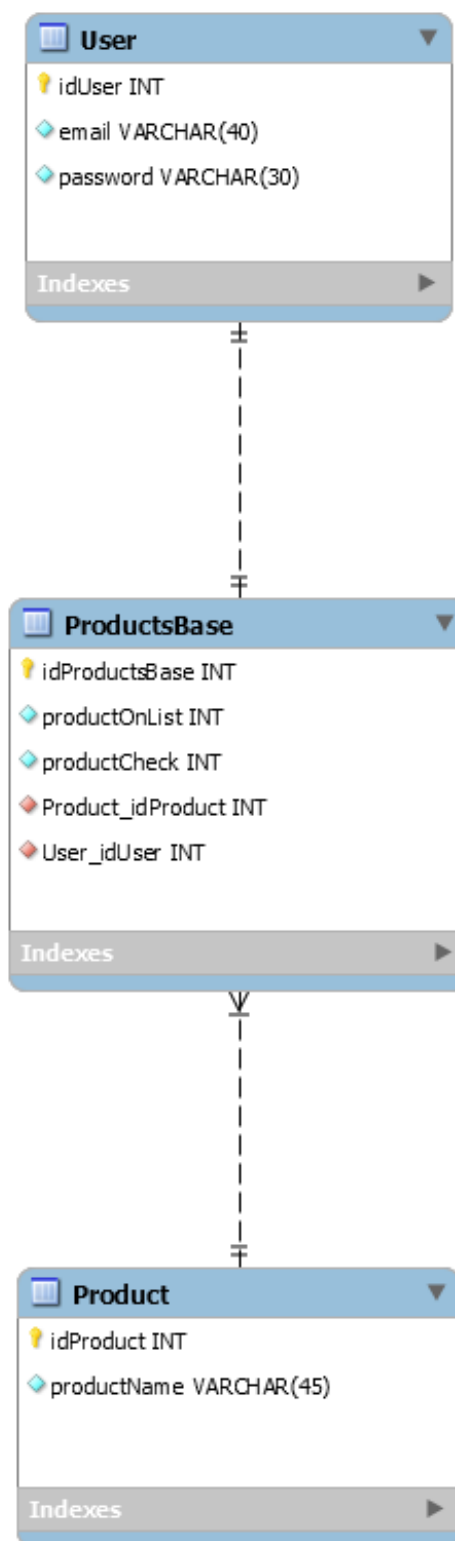


Diagram 1 Schemat bazy danych

### 3.2. Opis bazy danych

Diagram 1 przedstawia bazę danych, która zawiera informacje dotyczące użytkownika aplikacji oraz listę jego wszystkich produktów, które dodał. Schemat jest przedstawiony w formie graficznej utworzonej w programie MySQL Workbench. Baza składa się z trzech tabel.

Aplikacja wykorzystuje system zarządzania bazą danych 'MongoDB' z użyciem narzędzia 'Mongoose', które ułatwiło pracę dzięki schematom do modelowania danych aplikacji.

Baza danych umożliwia dostęp do informacji:

- Danych o koncie użytkownika – e-mail i hasło
- Nazwy produktów dodanych do bazy produktów użytkownika
- Status produktu – czy został dodany na aktualną listę zakupów
- Status produktu na aktualnej liście – czy został odhaczony

Założenia bazy danych:

- Indywidualna baza produktów dla każdego użytkownika
- Przechowywanie i zapisywanie stanu aktualnej listy zakupów użytkownika

Omówienie tabel:

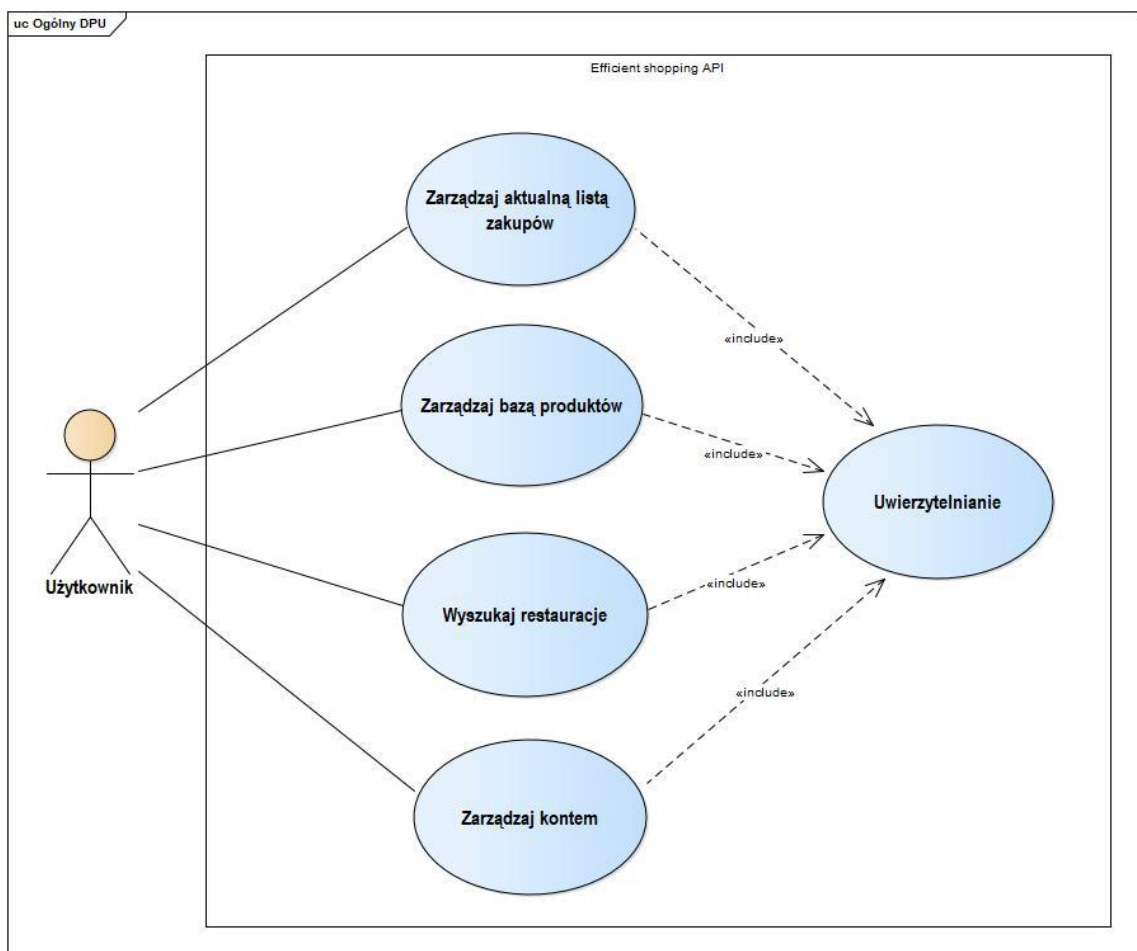
- *User* – tabela zawiera dane zarejestrowanych użytkowników w aplikacji – adres e-mail oraz hasło
- *ProductsBase* – tabela zawiera informacje o statusie produktu dodanego do bazy przez użytkownika – czy został dodany na aktualną listę zakupów
- *Product* – tabela zawiera dane o nazwie pojedynczego produktu

### 3.3. Diagram przypadków użycia

**Diagram przypadków użycia** opisuje wymagania systemu – jest graficznym (język UML) przedstawieniem przypadków użycia, czyli interakcji pomiędzy aktorem (użytkownikiem), który inicjuje zdarzenie oraz samym systemem jako sekwencję prostych kroków. Polega głównie na przedstawieniu systemu z punktu widzenia przyszłego użytkownika.

#### 3.3.1. Uogólniony diagram przypadków użycia

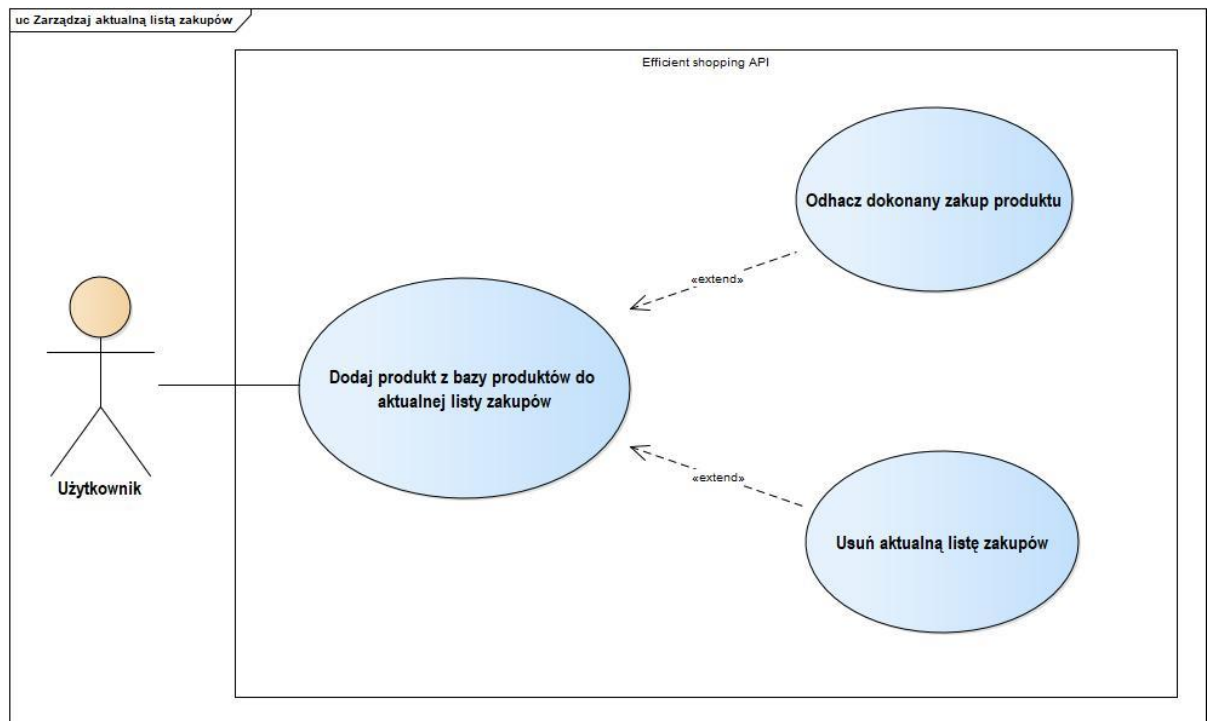
*Diagram przypadków użycia*



**Diagram 2 Uogólniony diagram przypadków użycia**

### 3.3.2. Rozwinięcie przypadku użycia „Zarządzaj aktualną listą zakupów”

*Diagram przypadków użycia*



**Diagram 3 Rozwinięcie przypadku użycia „Zarządzaj aktualną listą zakupów”**

**Scenariusz przypadku użycia „Dodaj produkt z bazy produktów do aktualnej listy zakupów”**

Nazwa	Dodaj produkt z bazy produktów do aktualnej listy zakupów
Numer	1
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może dodawać produkty ze swojej bazy produktów do aktualnej listy zakupów
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z aktualną listą zakupów
Warunki końcowe	Produkt został dodany do aktualnej listy zakupów
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1. Użytkownik dodaje wybrane przez siebie pozycję poprzez kliknięcie na nazwę produktu bądź przycisku symbolizującego dodanie (niebieski przycisk z plusem)</li> <li>2. Użytkownik widzi swoją zaktualizowaną listę zakupów na ekranie aktualnej listy zakupów oraz adnotację „Already in shopping list” na ekranie bazy produktów</li> </ol>
Alternatywne przepływy zdarzeń	1a. Aplikacja traci połączenie z internetem – system nie może nawiązać połączenia z bazą danych
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 1 Scenariusz przypadku użycia „Dodaj produkt z bazy produktów do aktualnej listy zakupów”**

**Scenariusz przypadku użycia „Odznacz dokonany zakup produktu”**

Nazwa	Odznacz dokonany zakup produktu
Numer	2
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może odhaczyć/zaznaczyć wybrany produkt, jako dokonany już zakup
Warunki wstępne	Użytkownik zalogował się do aplikacji, znajduje się na ekranie z aktualną listą zakupów oraz ma dodaną pozycję na listę
Warunki końcowe	Produkt został odznaczony/zaznaczony
Główny przepływ zdarzeń	1. Użytkownik klika na wybraną pozycję do odhaczenia 2. Użytkownik widzi zaktualizowany status produktu jako odznaczony („ptaszek”)
Alternatywne przepływy zdarzeń	1a.1. Użytkownik klika na wybraną pozycję, która jest odznaczona, w celu przywrócenia jej stanu 1a.2. Użytkownik widzi zaktualizowany status produktu jako nieodznaczony
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 2 Scenariusz przypadku użycia „Odznacz dokonany zakup produktu”**

**Scenariusz przypadku użycia „Usuń aktualną listę zakupów”**

Nazwa	Usuń aktualną listę zakupów
Numer	3
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może usunąć całą aktualną listę zakupów
Warunki wstępne	Użytkownik zalogował się do aplikacji, znajduje się na ekranie z aktualną listą zakupów oraz ma dodaną pozycję na listę
Warunki końcowe	Lista zakupów została usunięta
Główny przepływ zdarzeń	<ol style="list-style-type: none"><li>1. Użytkownik klika na przycisk usunięcia listy (czerwony przycisk z minusem)</li><li>2. Użytkownik potwierdza usunięcie listy zakupów</li><li>3. Użytkownik widzi zaktualizowaną – usuniętą - listę zakupów na ekranie aktualnej listy zakupów oraz zniknięcie adnotacji „Already in shopping list” przy usuniętych rekordach na ekranie bazy produktów</li></ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"><li>2a. Użytkownik anuluje usunięcie listy zakupów</li><li>3a. Użytkownik nie widzi aktualizacji listy, z powodu braku pozycji do usunięcia</li></ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 i 2 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 3 Scenariusz przypadku użycia „Usuń aktualną listę zakupów”**



### 3.3.3. Rozwinięcie przypadku użycia „Zarządzaj bazą produktów”

Diagram przypadków użycia

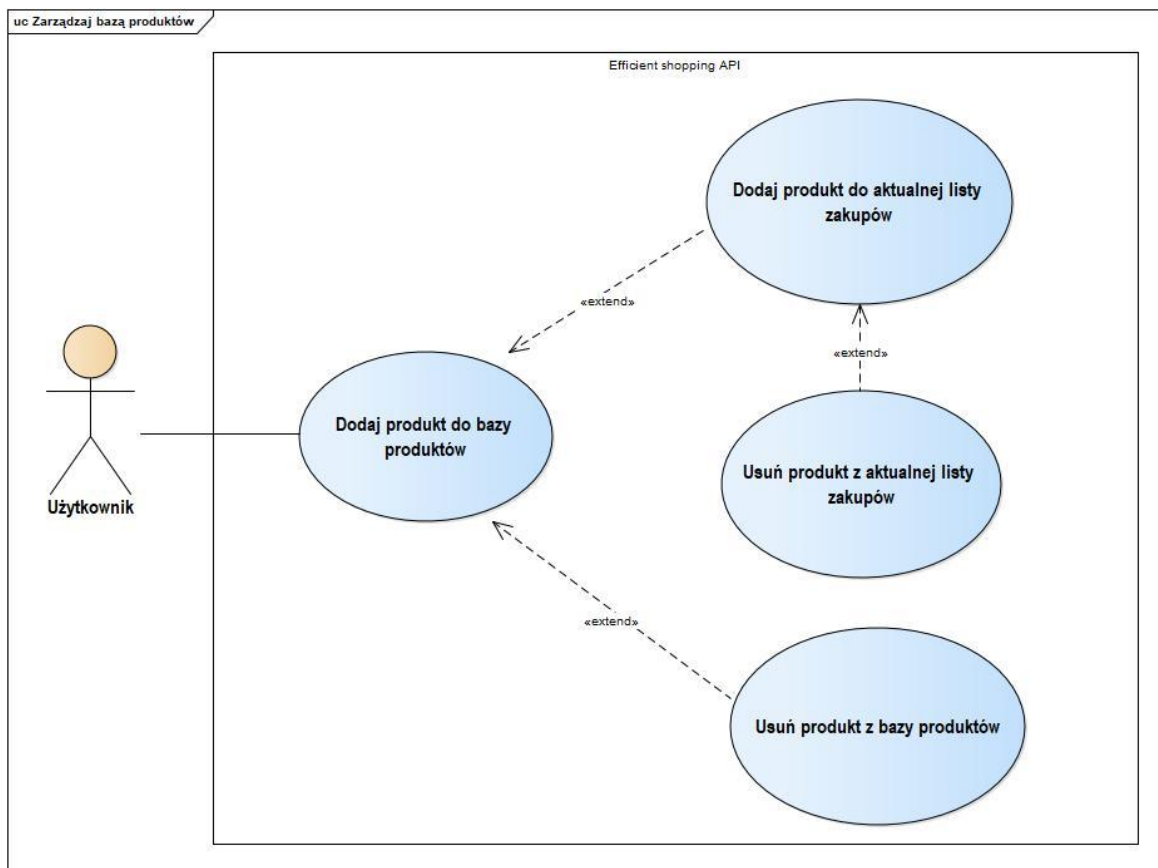


Diagram 4 Rozwinięcie przypadku użycia „Zarządzaj bazą produktów”

**Scenariusz przypadku użycia „Dodaj produkt do bazy produktów”**

Nazwa	Dodaj produkt do bazy produktów
Numer	1
Twórca	Przemysław Dzenisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może dodać nowy rekord do swojej bazy produktów
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z bazą produktów
Warunki końcowe	Nowy rekord został dodany do bazy produktów
Główny przepływ zdarzeń	<ol style="list-style-type: none"><li>1. Użytkownik klika na przycisk dodania produktu do bazy (niebieski przycisk z plusem)</li><li>2. Użytkownik wpisuje nazwę produktu</li><li>3. Użytkownik zatwierdza dodanie produktu</li><li>4. Użytkownik widzi zaktualizowaną bazę produktów z nowo dodanym rekordem</li></ol>
Alternatywne przepływy zdarzeń	3a. Użytkownik nie może zatwierdzić dodania produktu, ponieważ zostawił puste pole z nazwą produktu
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1, 2 i 3 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 4 Scenariusz przypadku użycia „Dodaj produkt do bazy produktów”**

**Scenariusz przypadku użycia „Dodaj produkt do aktualnej listy zakupów”**

Nazwa	Dodaj produkt do aktualnej listy zakupów
Numer	2
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może dodać produkt do aktualnej listy zakupów z poziomu ekranu bazy produktów
Warunki wstępne	Użytkownik zalogował się do aplikacji, znajduje się na ekranie z bazą produktów oraz posiada dodany produkt do bazy
Warunki końcowe	Produkt został dodany do aktualnej listy zakupów
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1. Użytkownik dodaje wybraną przez siebie pozycję poprzez kliknięcie na nazwę produktu bądź przycisku symbolizującego dodanie (niebieski przycisk z plusem)</li> <li>2. Użytkownik widzi aktualizację bazy produktów poprzez pojawienie się adnotacji „Already in shopping list” na ekranie bazy produktów przy wybranych przed chwilą przez użytkownika rekordach</li> <li>3. Użytkownik widzi swoją zaktualizowaną listę zakupów na ekranie aktualnej listy zakupów</li> </ol>
Alternatywne przebiegi zdarzeń	1a. Aplikacja traci połączenie z internetem – system nie może nawiązać połączenia z bazą danych
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 5 Scenariusz przypadku użycia „Dodaj produkt do aktualnej listy zakupów”**

**Scenariusz przypadku użycia „Usuń produkt z aktualnej listy zakupów”**

Nazwa	Usuń produkt z aktualnej listy zakupów
Numer	3
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może usunąć pojedynczy rekord z aktualnej listy zakupów z poziomu ekranu z bazą produktów
Warunki wstępne	Użytkownik zalogował się do aplikacji, znajduje się na ekranie z bazą produktów oraz posiada dodany produkt do aktualnej listy zakupów
Warunki końcowe	Produkt został usunięty z aktualnej listy zakupów
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1. Użytkownik klika na wybraną przez siebie pozycję - na nazwę produktu bądź przycisk symbolizujący dodanie (plus)</li> <li>2. Użytkownik widzi aktualizację bazy produktów poprzez zniknięcie adnotacji „Already in shopping list” na ekranie bazy produktów przy usuniętych pozycjach</li> <li>3. Użytkownik widzi swoją zaktualizowaną listę zakupów na ekranie aktualnej listy zakupów</li> </ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> <li>2. Aplikacja traci połączenie z internetem – system nie może nawiązać połączenia z bazą danych</li> </ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 6 Scenariusz przypadku użycia „Usuń produkt z aktualnej listy zakupów”**

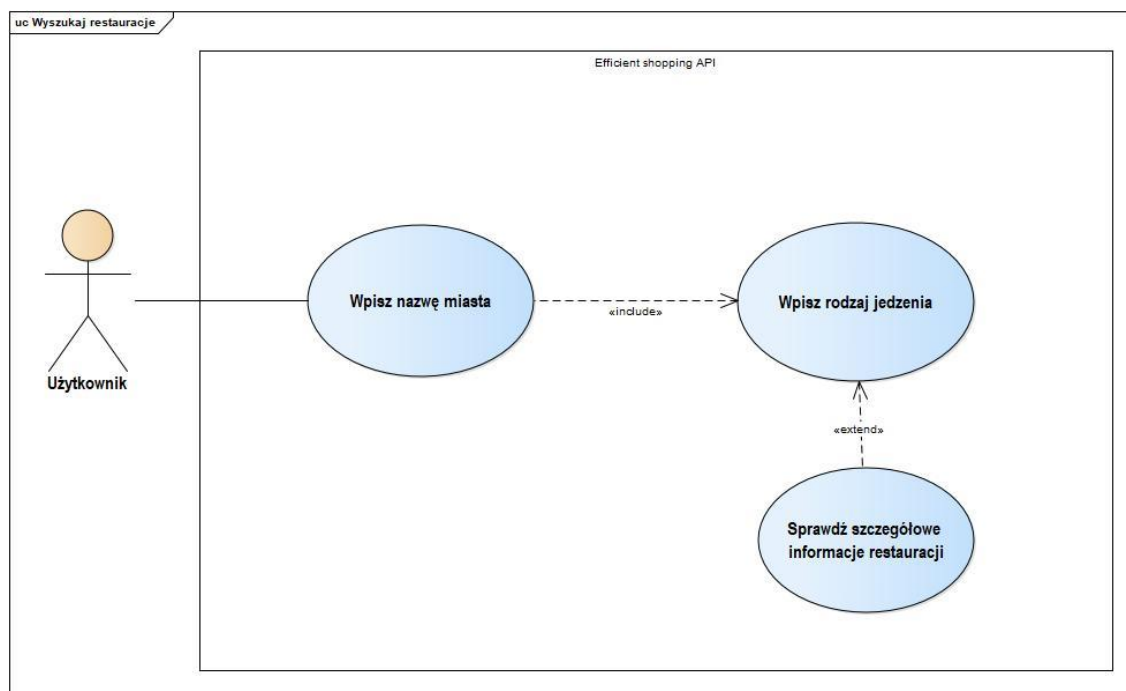
**Scenariusz przypadku użycia „Usuń produkt z bazy produktów”**

Nazwa	Usuń produkt z bazy produktów
Numer	4
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może usunąć dany rekord ze swojej bazy produktów na stałe
Warunki wstępne	Użytkownik zalogował się do aplikacji, znajduje się na ekranie z bazą produktów oraz posiada rekord w bazie produktów
Warunki końcowe	Rekord został usunięty na stałe z bazy produktów
Główny przepływ zdarzeń	<ol style="list-style-type: none"><li>1. Użytkownik klika na przycisk symbolizujący usunięcie rekordu (czerwony przycisk z minusem)</li><li>2. Użytkownik widzi zaktualizowaną listę na ekranie bazy produktów</li></ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"><li>1a. Aplikacja traci połączenie z internetem – system nie może nawiązać połączenia z bazą danych</li><li>2a. Użytkownik widzi zaktualizowaną listę na ekranie aktualnej listy zakupów – wywołanej przez usunięcie rekordu</li></ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 7 Scenariusz przypadku użycia „Usuń produkt z bazy produktów”**

### 3.3.4. Rozwinięcie przypadku użycia „Wyszukaj restauracje”

*Diagram przypadków użycia*



**Diagram 5 Rozwinięcie przypadku użycia „Wyszukaj restauracje”**

**Scenariusz przypadku użycia „Wpisz nazwę miasta”**

Nazwa	Wpisz nazwę miasta
Numer	1
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może wyszukać lokal gastronomiczny – musi w tym celu podać lokalizację
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z wyszukiwaniem restauracji
Warunki końcowe	Użytkownik podał lokalizację, potrzebne do wyszukania lokalów gastronomicznych
Główny przepływ zdarzeń	<ol style="list-style-type: none"><li>1. Użytkownik wpisuje lokalizację</li><li>2. Użytkownik zatwierdza klikając przycisk na klawiaturze bądź przechodząc dalej</li><li>3. Użytkownik widzi wyszukane wyniki przez aplikację</li></ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"><li>1a. Użytkownik zostawia puste pole z lokalizacją lub frazą jedzenia – aplikacja bierze pod uwagę tylko wypełnione pola i wyszukuje pierwsze 50 pozycji z Polski</li></ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	<ol style="list-style-type: none"><li>1. Użytkownik w 1 i 2 kroku może zakończyć (przerwać) działanie aplikacji</li><li>2. Aplikacja może pokazać wyniki po wypełnieniu tylko jednego pola</li></ol>

**Tabela 8 Scenariusz przypadku użycia „Wpisz nazwę miasta”**

**Scenariusz przypadku użycia „Wpisz rodzaj jedzenia”**

Nazwa	Wpisz rodzaj jedzenia
Numer	2
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może wyszukać lokal gastronomiczny – musi w tym celu podać frazę/rodzaj jedzenia
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z wyszukiwaniem restauracji
Warunki końcowe	Użytkownik podał rodzaj jedzenia, które aplikacja ma wziąć pod uwagę szukając lokalu gastronomicznego
Główny przepływ zdarzeń	<ol style="list-style-type: none"><li>1. Użytkownik wpisuje frazę/rodzaj jedzenia</li><li>2. Użytkownik zatwierdza klikając przycisk na klawiaturze bądź przechodząc dalej</li><li>3. Użytkownik widzi wyszukane wyniki przez aplikację</li></ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"><li>1a. Użytkownik zostawia puste pole z lokalizacją lub frazą jedzenia – aplikacja bierze pod uwagę tylko wypełnione pola i wyszukuje pierwsze 50 pozycji z Polski</li></ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	<ol style="list-style-type: none"><li>1. Użytkownik w 1 i 2 kroku może zakończyć (przerwać) działanie aplikacji</li><li>2. Aplikacja może pokazać wyniki po wypełnieniu tylko jednego pola</li></ol>

**Tabela 9 Scenariusz przypadku użycia „Wpisz rodzaj jedzenia”**



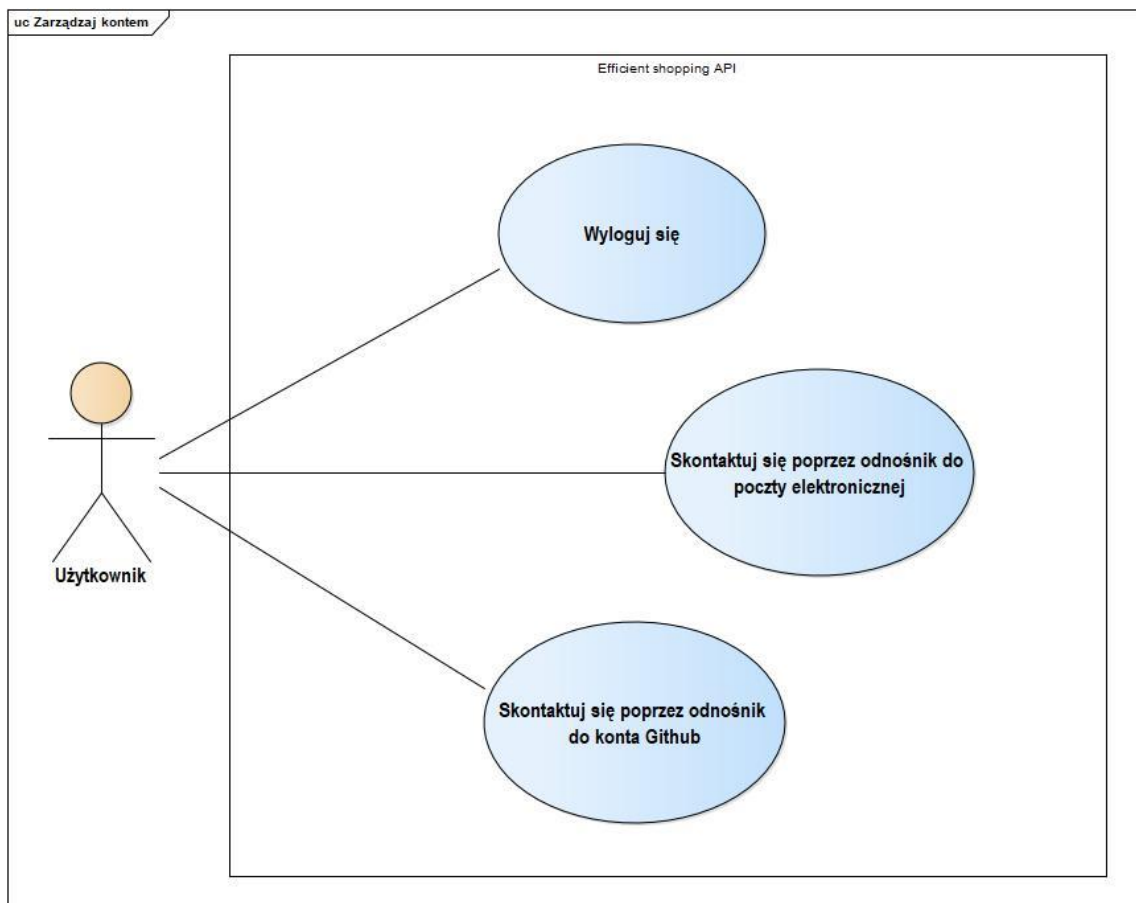
**Scenariusz przypadku użycia „Sprawdź szczegółowe informacje restauracji”**

Nazwa	Sprawdź szczegółowe informacje restauracji
Numer	3
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może dowiedzieć się więcej o danym lokalu gastronomicznym
Warunki wstępne	Użytkownik zalogował się do aplikacji, znajduje się na ekranie z wyszukiwaniem restauracji oraz wypełnił pola wyszukiwania lokalu względem swoich potrzeb
Warunki końcowe	Użytkownik otrzymał szczegółowe informacje o wybranym przez siebie lokalu gastronomicznym
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1. Użytkownik klika na wybrany przez siebie lokal gastronomiczny – na zdjęcie lub nazwę lokalu</li> <li>2. Użytkownik zostaje przeniesiony na ekran ze szczegółowymi informacjami o lokalu gastronomicznym</li> </ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> <li>1a. Aplikacja traci połączenie z internetem – system nie może nawiązać połączenia z bazą danych lokali</li> <li>2a. Dany lokal nie ma uzupełnionych danych informacyjnych</li> </ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 i 2 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 10 Scenariusz przypadku użycia „Sprawdź szczegółowe informacje restauracji”**

### 3.3.5. Rozwinięcie przypadku użycia „Zarządzaj kontem”

*Diagram przypadków użycia*



**Diagram 6 Rozwinięcie przypadku użycia „Zarządzaj kontem”**

### **Scenariusz przypadku użycia „Wyloguj się”**

Nazwa	Wyloguj się
Numer	1
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może wylogować się ze swojego konta w aplikacji
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z zarządzania kontem
Warunki końcowe	Użytkownik został wylogowany z konta w aplikacji
Główny przepływ zdarzeń	<ol style="list-style-type: none"><li>1. Użytkownik klika na przycisk wylogowania się z konta – przycisk z napisem „Sign out”</li><li>2. Użytkownik został wylogowany z konta – przeniesiony na ekran logowania się do aplikacji</li></ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 i 2 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 11 Scenariusz przypadku użycia „Wyloguj się”**

**Scenariusz przypadku użycia „Skontaktuj się poprzez odnośnik do poczty elektronicznej”**

Nazwa	Skontaktuj się poprzez odnośnik do poczty elektronicznej
Numer	2
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może skontaktować się z autorem aplikacji bezpośrednio poprzez odnośnik z adresem poczty elektronicznej
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z zarządzania kontem
Warunki końcowe	Użytkownik został przeniesiony na swoją pocztę elektroniczną z formularzem wysyłania wiadomości
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1. Użytkownik klika na odnośnik z nazwą poczty elektronicznej autora</li> <li>2. Użytkownik zostaje przeniesiony na zewnętrzną stronę/aplikację ze swoją pocztą elektroniczną</li> </ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> <li>1a. Błędnie podana nazwa poczty elektronicznej</li> <li>2a. Brak zezwolenia na używanie zewnętrznych stron/aplikacji</li> </ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 12 Scenariusz przypadku użycia „Skontaktuj się poprzez odnośnik do poczty elektronicznej”**

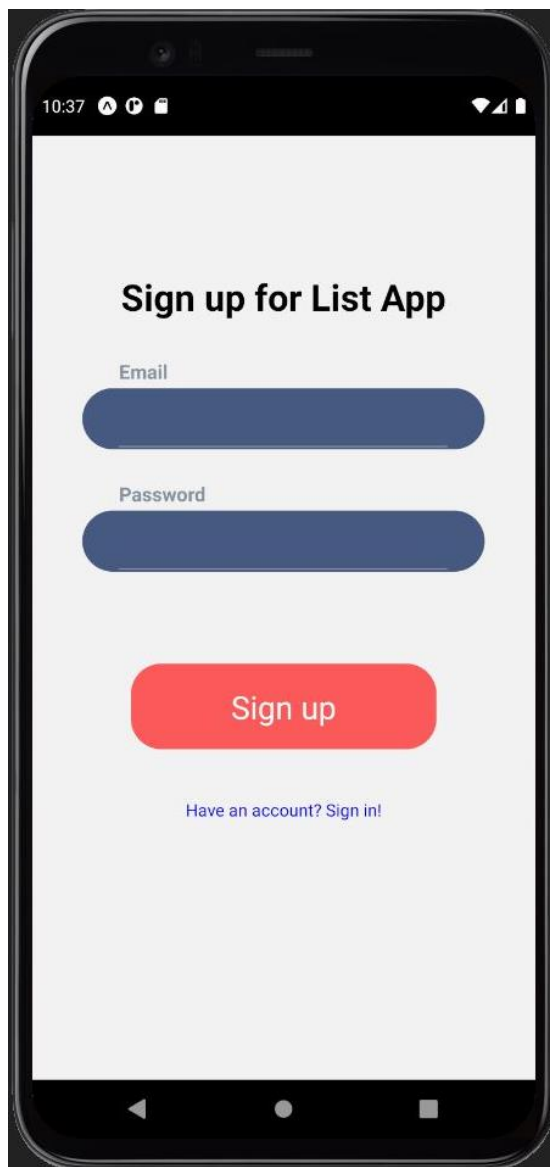
**Scenariusz przypadku użycia „Skontaktuj się poprzez odnośnik do konta Github”**

Nazwa	Skontaktuj się poprzez odnośnik do konta Github
Numer	3
Twórca	Przemysław Dzienisiewicz
Poziom ważności	Istotny
Typ przypadku użycia	Ogólny
Aktorzy	Użytkownik
Krótki opis	Użytkownik może skontaktować się z autorem aplikacji bezpośrednio poprzez odnośnik do konta Github
Warunki wstępne	Użytkownik zalogował się do aplikacji oraz znajduje się na ekranie z zarządzania kontem
Warunki końcowe	Użytkownik został przeniesiony na stronę z kontem Github autora aplikacji
Główny przepływ zdarzeń	<ol style="list-style-type: none"> <li>1. Użytkownik klika na odnośnik z nazwą poczty elektronicznej autora</li> <li>2. Użytkownik zostaje przeniesiony na zewnętrzną stronę z kontem Github autora aplikacji</li> </ol>
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> <li>1a. Błędny link konta Github</li> <li>2a. Brak zezwolenia na używanie zewnętrznych stron</li> </ol>
Specjalne wymagania	Aplikacja ma działać w czasie rzeczywistym
Notatki i kwestie	Użytkownik w 1 kroku może zakończyć (przerwać) działanie aplikacji

**Tabela 13 Scenariusz przypadku użycia „Skontaktuj się poprzez odnośnik do konta Github”**

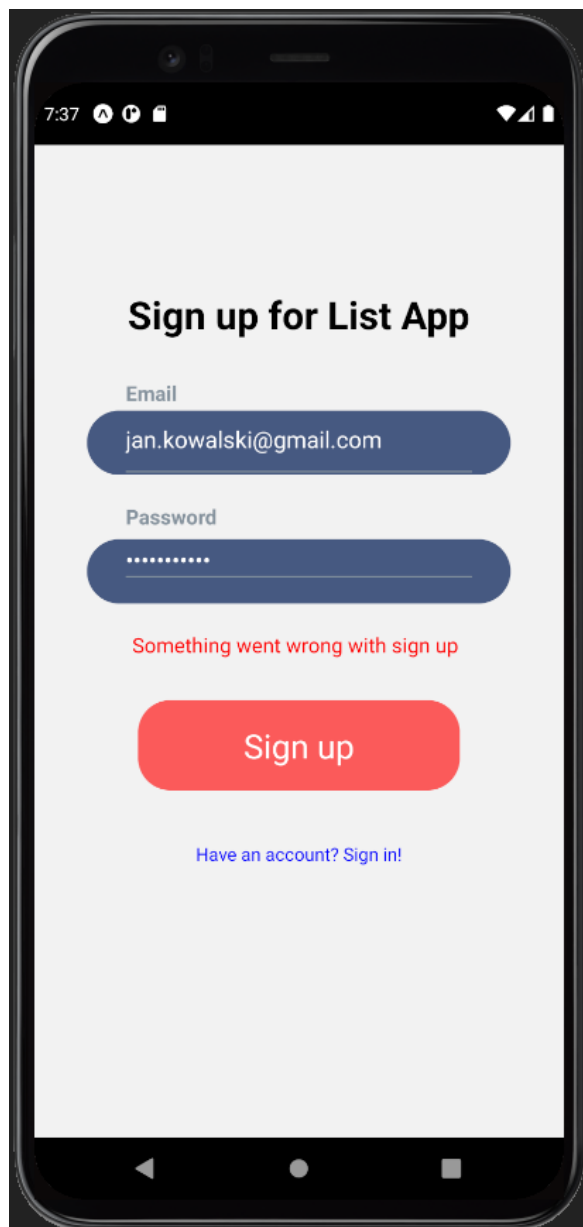
## 4. Dokumentacja użytkownika

### *Ekran rejestracji do aplikacji*



**Rysunek 1 Ekran rejestracji**

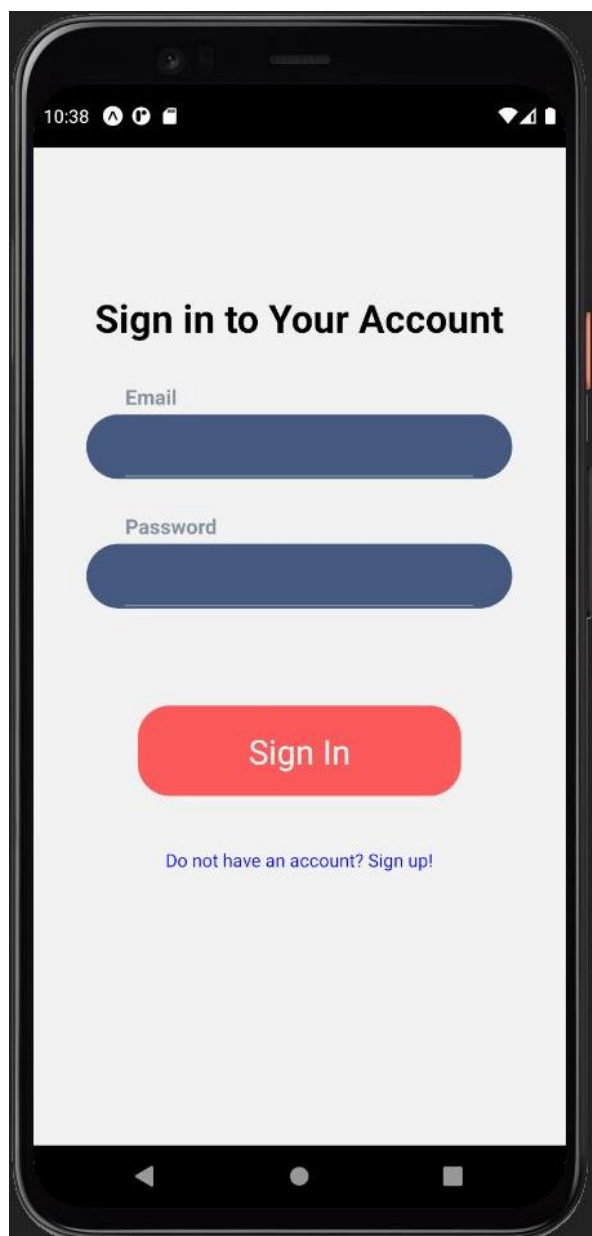
Po włączeniu aplikacji na swoim urządzeniu mobilnym, użytkownik zobaczy ekran z panelem rejestracyjnym nowego konta - przedstawiony na Rysunek 1. Po podaniu prawidłowych danych – nazwy e-mail oraz hasła, na które nie jest zarejestrowane jeszcze żadne konto – użytkownik zostanie bezpośrednio przeniesiony na ekran z panelem swojej aktualnej listy zakupów.



**Rysunek 2 Ekran rejestracji z błędem**

W przypadku błędnie podanych danych do rejestracji nowego konta – użytkownik zobaczy stosowny, czerwony komunikat na ekranie, tj. „Something went wrong with sign up” – przedstawiony na Rysunek 2.

## *Ekran logowania do aplikacji*



**Rysunek 3 Ekran logowania**

W przypadku, gdy użytkownik posiada już konto w aplikacji – może przejść bezpośrednio do ekranu z panelem logowania się - Rysunek 3 – poprzez kliknięcie niebieskiego odnośnika z napisem „Have an account? Sign in!” – przedstawionego na Rysunek 1 oraz Rysunek 2.

Analogicznie użytkownik może wrócić z ekranu logowania na ekran rejestracji konta poprzez kliknięcie niebieskiego odnośnika z napisem „Do not have an account? Sign up!” – przedstawionego na Rysunek 3.

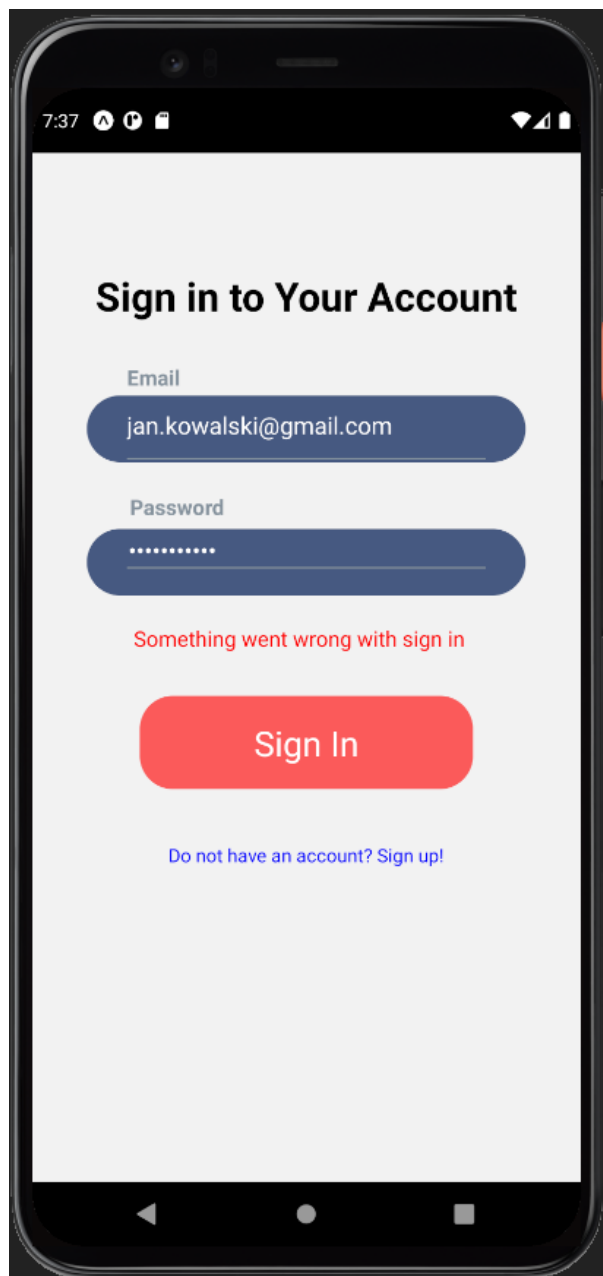


W aplikacji został zainicjowany także asynchroniczny system przechowywania klucza i wartości, który jest globalny dla aplikacji – została do tego użyta biblioteka 'Async-storage'.

Pozwala on na zaimplementowanie rozwiązania, dzięki któremu użytkownik nie musi logować się ponownie za każdym razem uruchamiania aplikacji – stan logowania zostaje przechowywany lokalnie na urządzeniu mobilnym, do czasu samodzielnego wylogowania się użytkownika w aplikacji. Implementacja opisanej metody została przedstawiona na Rysunek 4.

```
20
21  const tryLocalSignin = (dispatch) => async () => {
22    const token = await AsyncStorage.getItem("token");
23    if (token) {
24      dispatch({ type: "signin", payload: token });
25      navigate("Products");
26    } else {
27      navigate("Signup");
28    }
29  };
30
```

Rysunek 4 Metoda przechowywania klucza lokalnie w aplikacji



**Rysunek 5 Ekran logowania z błędem**

Po wpisaniu prawidłowych danych logowania do swojego konta – tj. e-mail i hasło – użytkownik zostanie bezpośrednio przeniesiony na ekran z panelem swojej aktualnej listy zakupów – przedstawiony na Rysunek 6.

W przypadku błędnie wpisanych danych logowania – użytkownik zobaczy stosowny, czerwony komunikat na ekranie, tj. „Something went wrong with sign in” – przedstawiony na Rysunek 5.

### *Ekran aktualnej listy zakupów – pusta lista*



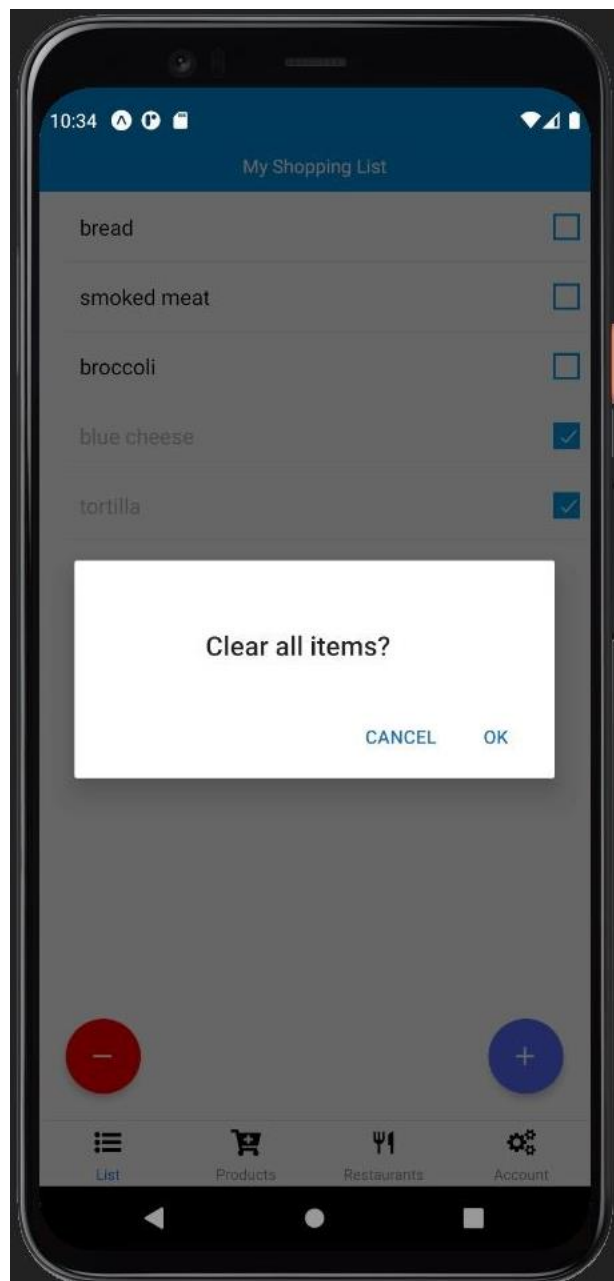
**Rysunek 6 Ekran aktualnej listy zakupów – pusta lista**

Po poprawnym procesie rejestracji konta bądź logowania, użytkownik zostanie przeniesiony na ekran z panelem aktualnej listy zakupów, widocznego na Rysunek 6. Dla nowo powstałego konta – baza produktów, co za tym idzie – także aktualna lista produktów – jest pusta. Na górze ekranu, pod samym panelem powiadomień swojego urządzenia mobilnego, widnieje podpis, na którym ekranie użytkownik aktualnie się znajduje – tj. „My Shopping List”.

U dołu ekranu znajdują się dwa przyciski – usuwania (czerwony przycisk z minusem) omówiony w sekcji Usuwanie Aktualnej Listy Produktów – oraz dodawania produktów (niebieski przycisk z plusem), który po naciśnięciu bezpośrednio przekieruje użytkownika na ekran z panelem bazy produktów – przedstawionego na Rysunek 8.

Innym elementem u dołu rzucającym się oczom użytkownika – jest nawigacyjny pasek kart, który umożliwia łatwe przełączanie się między różnymi ekranami aplikacji – pasek został wskazany czerwonym obramowaniem na Rysunek 6.

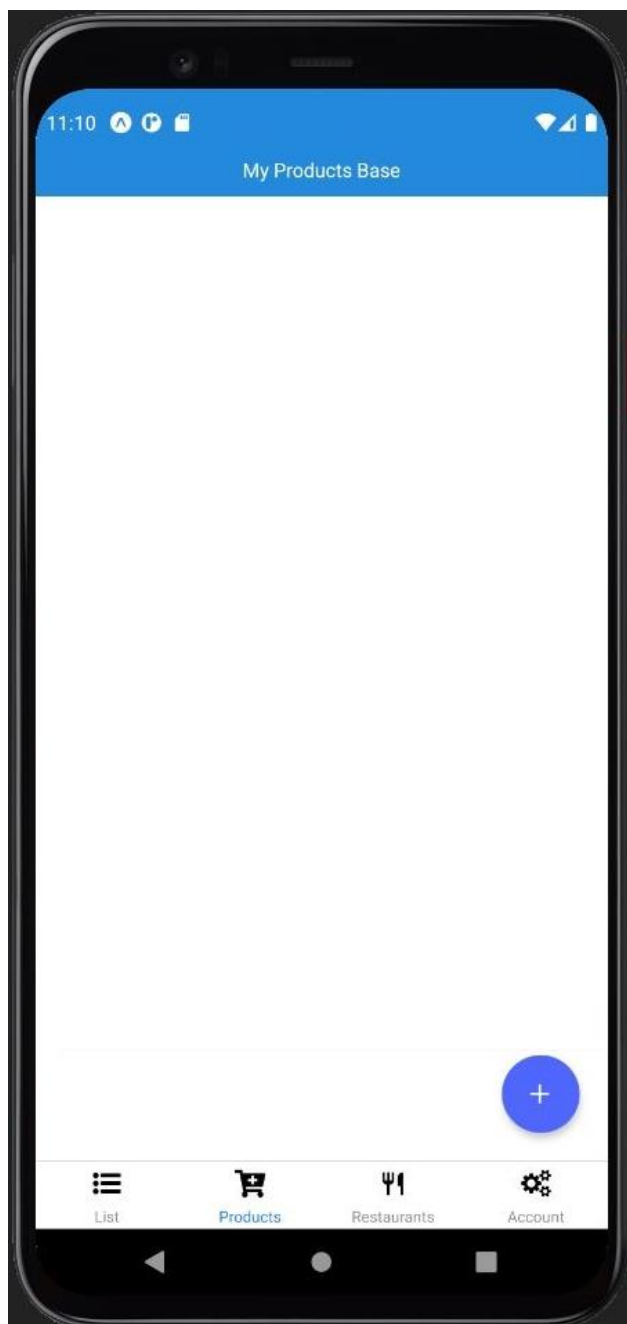
Znajdują się na nim cztery, podpisane obrazy, odpowiadające poszczególnym ekranom po naciśnięciu przez użytkownika. Przełączanie się między ekranami jest inicjalizowane 'leniwie', tzn. że komponenty ekranu nie są montowane, dopóki nie zostaną najpierw skupione (wywołanie poprzez kliknięcie).



**Rysunek 7 Ekran usuwania całej aktualnej listy zakupów**

Na ekranie z panelem aktualnej listy zakupów – u dołu ekranu po lewej stronie znajduje się czerwony przycisk z minusem, który służy do usunięcia wszystkich pozycji z aktualnej listy zakupów użytkownika – tzn. ekranu „My Shopping List”. Po naciśnięciu przycisku ukaże się wyskakujące okienko ‘pop-up’, które zapyta o potwierdzenie bądź anulowanie usunięcia wszystkich pozycji z listy – przedstawione na Rysunek 7.

### *Ekran bazy produktów użytkownika – pusta lista*



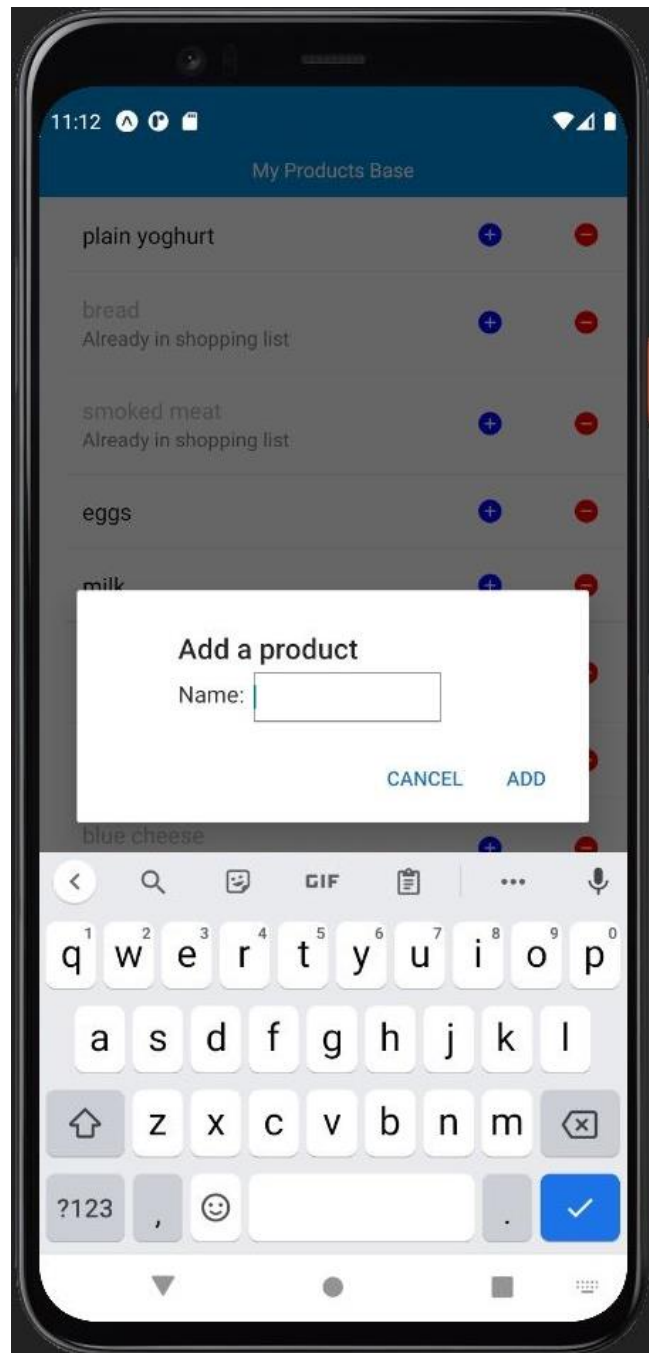
**Rysunek 8 Ekran bazy produktów – pusta lista**

Użytkownik może bezpośrednio przejść na ekran z panelem swojej bazy produktów po naciśnięciu niebieskiego przycisku z plusem, będąc na ekranie z panelem aktualnej listy zakupów – pokazanego m.in. na Rysunek 6 – lub po naciśnięciu ikony koszyka podpisanej 'Products' znajdującego się na pasku nawigacyjnym u dołu ekranu.

Analogicznie jak w panelu z ekranem aktualnej listy zakupów, nowy użytkownik początkowo posiada pustą bazę produktów.

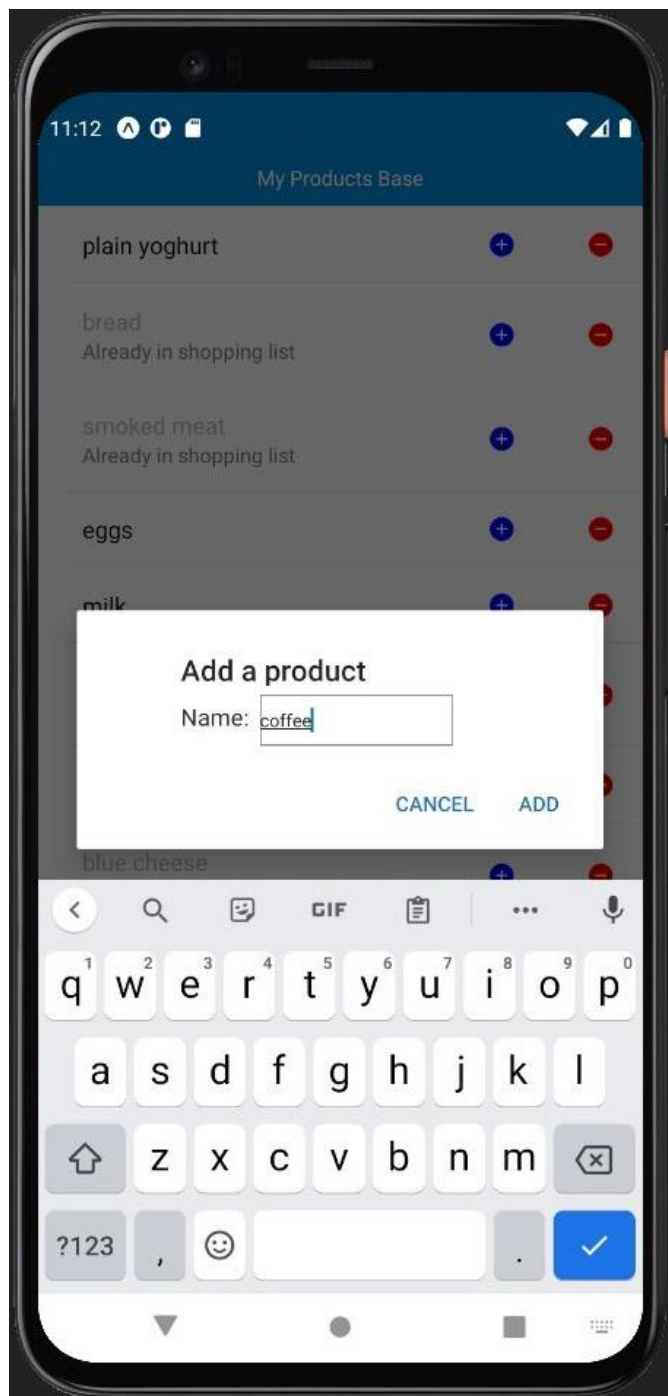
Na górze ekranu, pod panelem powiadomień urządzenia, widnieje podpis, na którym ekranie użytkownik aktualnie się znajduje – tj. „My Products Base”.

U dołu ekranu znajduje się niebieski przycisk z plusem, odpowiadający za dodawanie produktów do bazy - po naciśnięciu ukaże się wyskakujące okienko ‘pop-up’, w którym po wpisaniu nazwy swojego produktu i zatwierdzeniu przyciskiem ‘Add’ – dodaje nową pozycję na dół listy – proces dodawania produktów krok po kroku pokazany jest na Rysunek 9, Rysunek 10 oraz Rysunek 11.



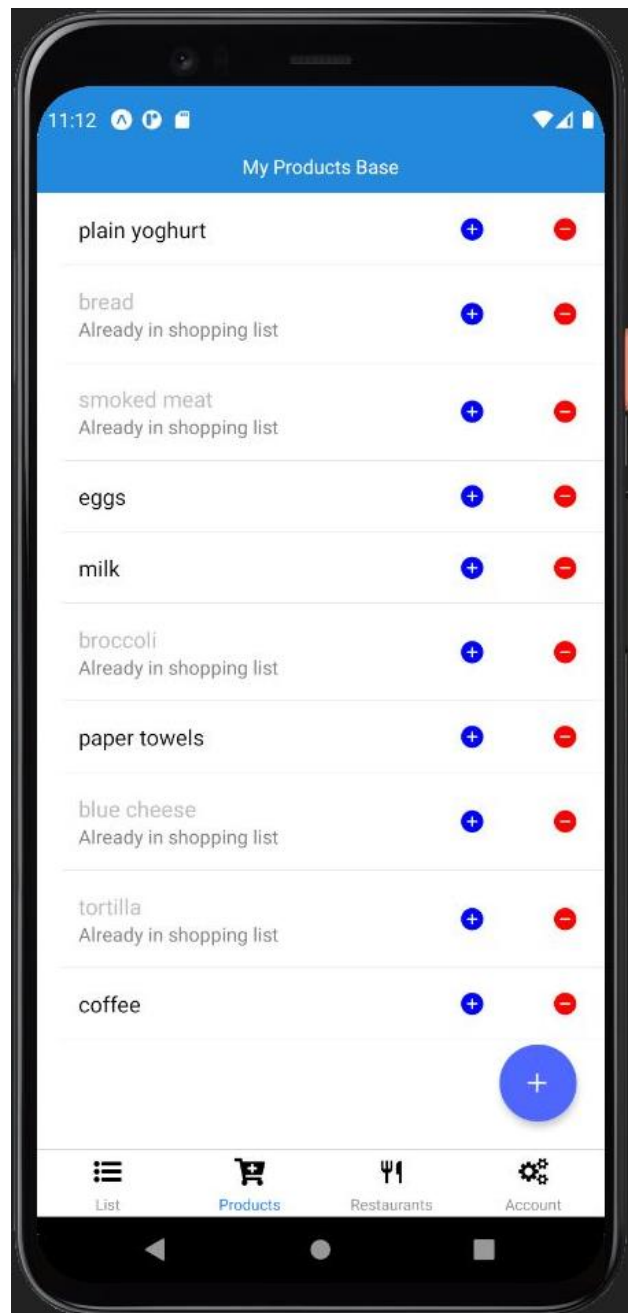
Rysunek 9 Ekran dodawania produktu do bazy





Rysunek 10 Ekran dodawania produktu do bazy – przykładowo uzupełniona nazwa produktu

*Ekran bazy produktów użytkownika – uzupełniona lista*



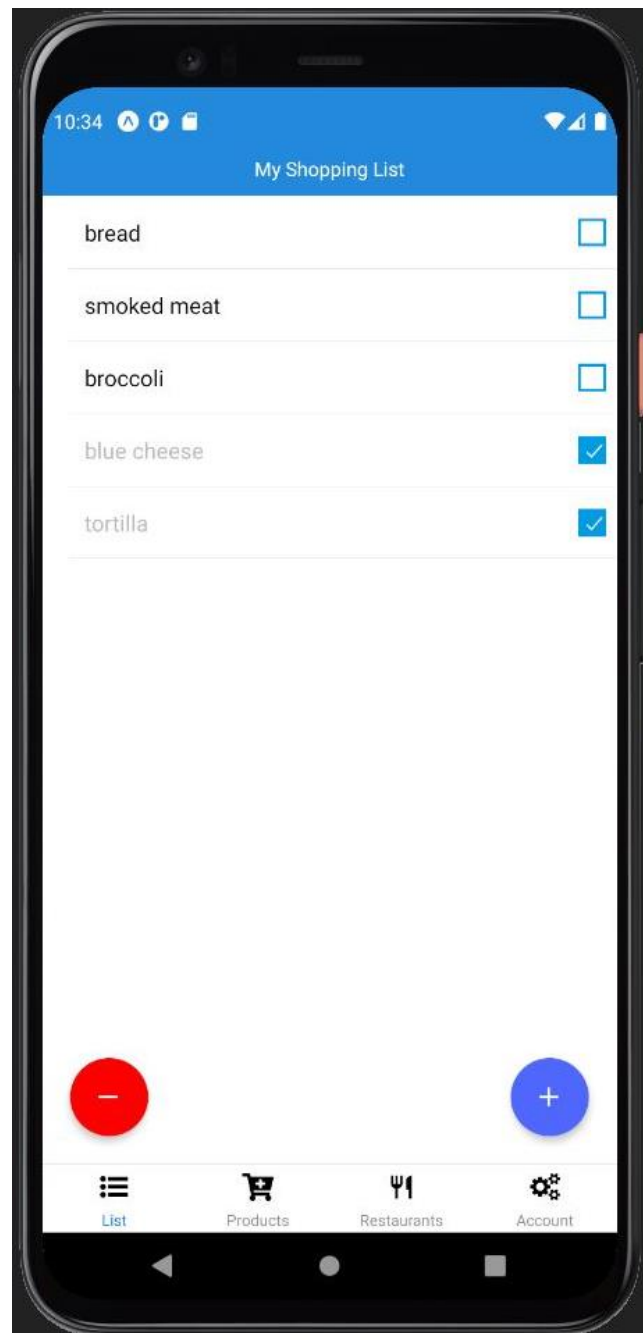
**Rysunek 11 Ekran bazy produktów – przykładowo uzupełniona lista**

Po dodaniu produktu do swojej bazy – użytkownik może teraz wybrać, które z nich chce umieścić na swojej aktualnej liście zakupów – pierwszym, głównym ekranie aplikacji. Może to zrobić poprzez kliknięcie na okrągły, niebieski przycisk z plusem obok produktu lub po prostu kliknąć na pozycję z nazwą produktu.

Po dodaniu produktu na aktualną listę, kolor nazwy pozycji zmieni swój kolor na szary oraz pojawi się pod nim adnotacja, tj. „Already in shopping list” – od teraz produkt jest na aktualnej liście zakupów.

Aby usunąć dany produkt z listy, wystarczy jeszcze raz kliknąć na dany artykuł. Obok nazwy produktu znajduje się jeszcze jeden przycisk – okrągły, czerwony z minusem – który służy do usuwania pozycji z bazy produktów użytkownika. Wszystkie omówione rzeczy zawarte są na Rysunek 11.

*Ekran aktualnej listy zakupów – uzupełniona lista*

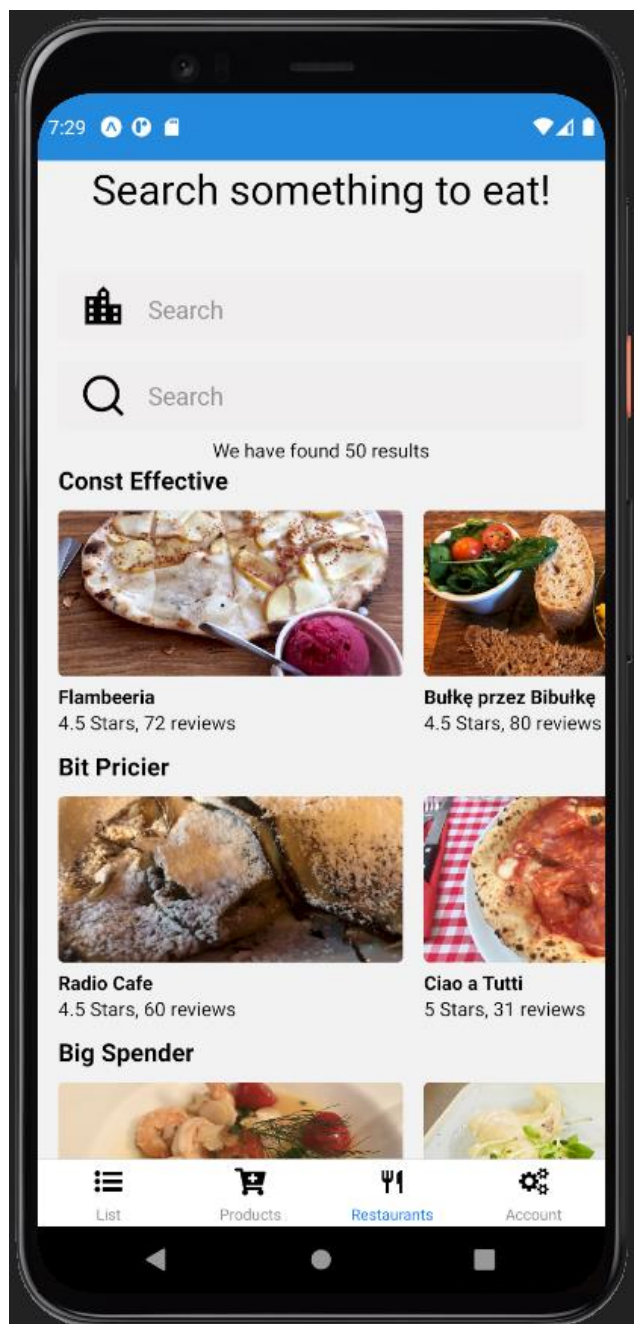


**Rysunek 12 Ekran aktualnej listy zakupów – przykładowo uzupełniona lista**

Po dodaniu produktów do bazy i wybraniu, które z nich chce umieścić na liście aktualnych zakupów – użytkownik może wrócić z powrotem na ekran z panelem aktualnych zakupów - poprzez kliknięcie ikony listy podpisanej 'List' znajdującej się na pasku nawigacyjnym u dołu ekranu. Widoczna będzie lista z produktami, które przed chwilą dodał.

Aby ułatwić zakupy, można łatwo odhaczać, które z produktów zostały już zakupione lub włożone do koszyka – obok nazwy pozycji znajduje się kwadratowy, dwustanowy przycisk wyboru - 'checkbox' - naciśnięcie na wspomniane okienko bądź po prostu na nazwę artykułu, skutkuje iż kolor produktu staje się bardziej przezroczysty (stopień widoczności elementu) oraz przycisk wyboru zmienia stan na zaznaczony („ptaszek”) z wypełnionym niebieskim kolorem – efekt został zaprezentowany na Rysunek 12.

## Ekran wyszukiwania restauracji

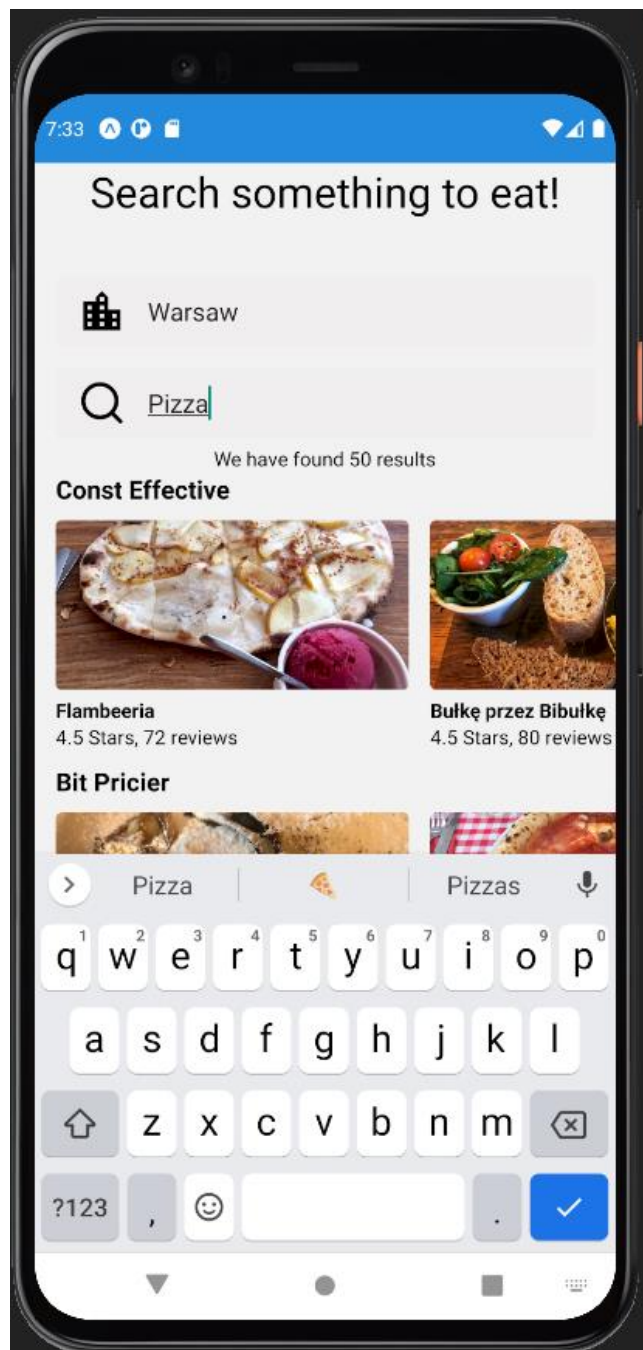


Rysunek 13 Ekran wyszukiwania restauracji – bez uzupełnionych pól wyszukiwania

Klikając na ikonę noża i widelca podpisanej 'Restaurants' znajdującej się na pasku nawigacyjnym u dołu ekranu, użytkownik zostanie przeniesiony na ekran z panelem wyszukiwania restauracji.

Znajduje się tutaj dodatkowa funkcjonalność aplikacji, dzięki której w przypadku chęci skorzystania z jedzenia na wynos – istnieje możliwość wyszukania pożądaney restauracji i otrzymania niezbędnych informacji do złożenia zamówienia.

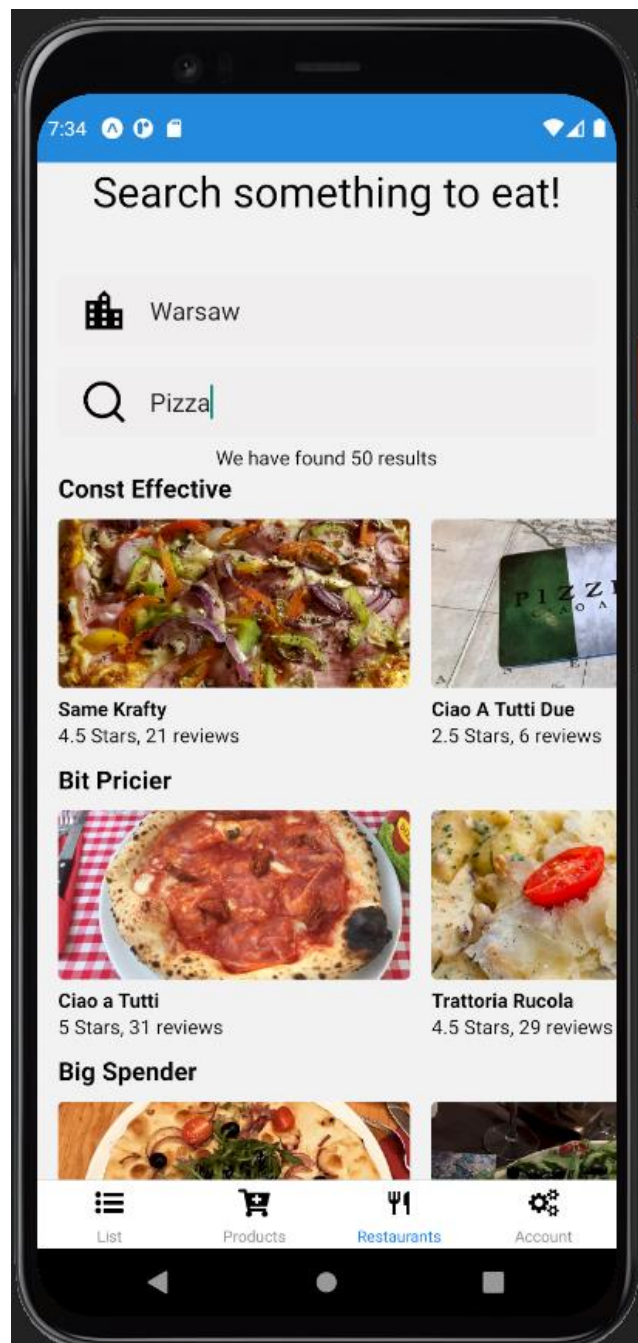
Podzielone jest to na dwa segmenty – pierwszy, w którym użytkownik wpisuje niezbędne informacje oraz drugi, gdzie otrzymuje wyniki wyszukiwania z nazwą, oceną oraz liczbą recenzji miejsca, które dodatkowo są podzielone na trzy kategorie cenowe – od tańszych do droższych lokali gastronomicznych – zaprezentowane na Rysunek 13.



**Rysunek 14 Ekran wyszukiwania restauracji – wypełnione przykładowo pola**

W pierwszym polu z ikoną budynków trzeba wpisać lokalizację – miasto – w którym chcemy znaleźć lokal gastronomiczny, następnie w polu z ikoną lupy wpisać frazę jedzenia, na które użytkownik ma ochotę, np. pizza, burger – przykładowo uzupełnione pola zaprezentowano na Rysunek 14.

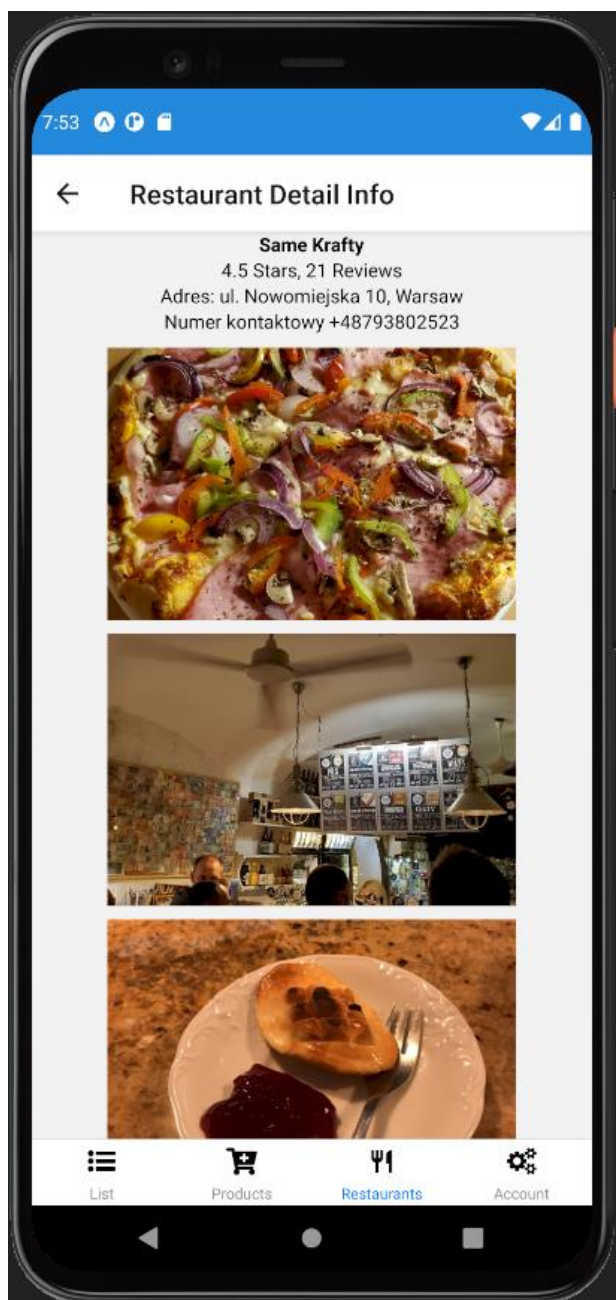




**Rysunek 15** Ekran wyszukiwania restauracji – przykładowa lista wyników

Po próbie znalezienia najbardziej dopasowanych miejsc względem naszych danych, zostajemy poinformowani małą adnotacją o liczbie wyników, np. „We have found 50 results” (pięćdziesiąt jest maksymalną liczbą na jedno wyszukiwanie) – przykładowo znalezione wyniki lokali gastronomicznych oraz adnotacja znajdują się na Rysunek 15.

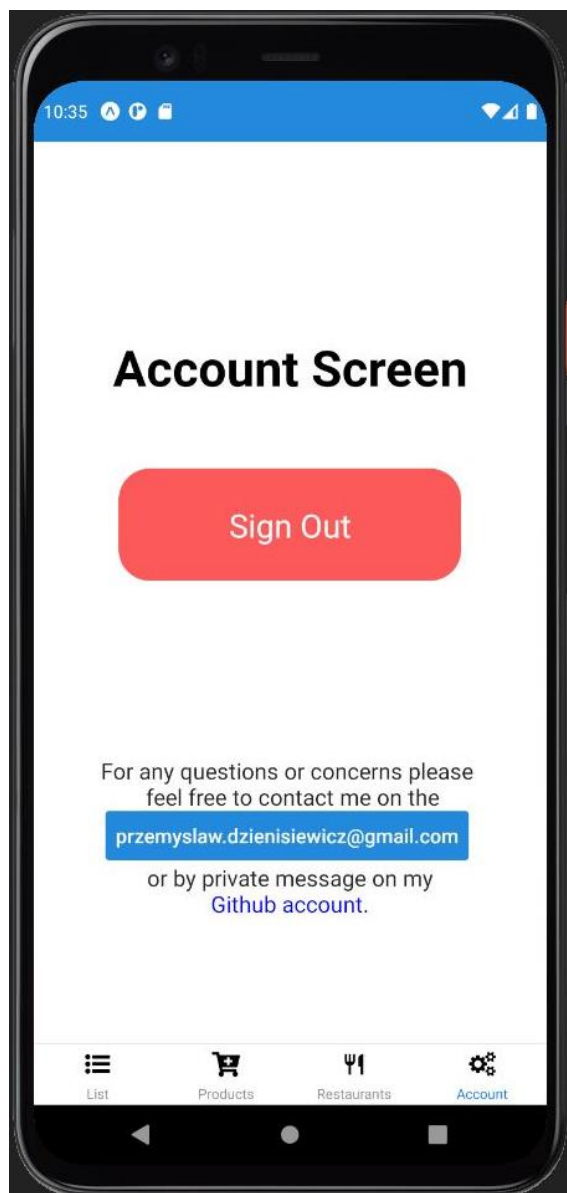
*Ekran wyszukiwania restauracji – ekran dodatkowych informacji o restauracji*



**Rysunek 16 Ekran wyszukiwania restauracji – przykładowy ekran dodatkowych informacji o wybranej restauracji**

Po wpisaniu i wyszukaniu pożądanego lokalu gastronomicznego, użytkownik może dowiedzieć się więcej na temat restauracji. Aby otrzymać dodatkowe informacje, trzeba kliknąć na wybrany wynik zdjęcia z nazwą lokalu. Użytkownik zostanie przeniesiony na ekran z szczegółowymi informacjami, tj. numer telefonu, adres, dodatkowe zdjęcia – przedstawione na Rysunek 16.

## Ekran zarządzania kontem



**Rysunek 17 Ekran zarządzania kontem**

Klikając na ostatnią ikonę znajdującą się na pasku nawigacyjnym u dołu ekranu - przedstawiającą przekładnię/koła zębate podpisaną 'Account', użytkownik zostanie przeniesiony na ekran z panelem zarządzania kontem.

Znajduje się tutaj czerwony przycisk do wylogowania się z konta – podpisany „Sign Out” oraz informacje kontaktowe z autorem aplikacji mobilnej w przypadku zgłaszania usterek technicznych bądź pytań – zawierają bezpośrednie linki do poczty elektronicznej oraz konta github – omawiany ekran znajduje się na Rysunek 17.

## 5. Podsumowanie

Celem stworzenia aplikacji, która jest tematem tejże pracy inżynierskiej, było usprawnienie robienia zakupów. Użytkownik aplikacji jest w stanie znacząco przyspieszyć swoje zakupy dzięki zaplanowaniu potrzebnych mu produktów i wprowadzeniu ich na listę w aplikacji przed wyruszeniem do sklepu. Aby zakupy przebiegały jeszcze sprawniej, w trakcie może odznaczać zakupione już artykuły.

Osobiście było to dla mnie dużym wyzwaniem, ponieważ był to pierwszy samodzielny projekt - od początku do końca - z wykorzystaniem biblioteki 'React Native'. Ostatecznie jestem zadowolony z wyniku pracy oraz tego jak dużo doświadczenia wniosła w moją przyszłą karierę programisty.

## **Streszczenie**

Celem pracy inżynierskiej jest stworzenie aplikacji mobilnej do sprawniejszego robienia zakupów, która będzie działać na obu systemach operacyjnych – android oraz iOS. Ma spełniać podstawowe zadanie, czyli umożliwić użytkownikowi skrócenie robienia zakupów - poprzez zaplanowanie jakie produkty chce kupić, wprowadzeniem ich do aplikacji na listę przed udaniem się do sklepu oraz odznaczanie zakupionych już artykułów.

## **Abstract**

The aim of the engineering work is to create a mobile application for more efficient shopping, which will work on both operating systems - Android and iOS. It is supposed to fulfill the basic task, in other words to enable the user to shorten the shopping process - by planning what products he wants to buy, entering them into the application on the list before going to the store and checking off already purchased items.

## Bibliografia

- [1] "React and React Native" Adam Boduch, Packt Publishing (March 8, 2017)
- [2] "React Native Cookbook – Second Edition" Dan Ward, Packt Publishing, 2nd Edition (January 31, 2019)
- [3] "React Native for Mobile Development" Akshat Paul and Abhishek Nalwaya, Apress, 2nd ed. edition (June 13, 2019)
- [4] "JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native and Electron 1st Edition" Adam D. Scott, O'Reilly Media, 1st edition (February 25, 2020)
- [5] "React and React Native—A complete hands-on guide to modern web and mobile development with React.js, 3rd Edition" Adam Boduch and Roy Derks, Packt Publishing (April 30, 2020)
- [6] <https://reactnative.dev/docs/getting-started>
- [7] <https://docs.expo.io>
- [8] <https://developer.mozilla.org/en-US/docs/Glossary/JavaScript>
- [9] <https://developer.mozilla.org/en-US/docs/Glossary/HTML>
- [10] <https://www.npmjs.com/package/nodemon>
- [11] <https://pl.wikipedia.org/wiki/MongoDB>
- [12] [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)
- [13] <https://ngrok.com/docs>
- [14] <https://www.atlassian.com/git/tutorials/git-bash>
- [15] <https://www.yelp.pl/about>

[16] <https://mongoosejs.com/>

[17] [https://en.wikipedia.org/wiki/Use\\_case](https://en.wikipedia.org/wiki/Use_case)

[18] <https://guides.github.com/activities/hello-world/>

[19] <https://reactjs.org/docs/introducing-jsx.html>

[20] <https://help.trello.com/article/708-what-is-trello>

[21] <https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.1>

[22] <https://developer.android.com/studio/intro>

[23] [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics)

## Spis Tabel

Tabela 1 Scenariusz przypadku użycia „Dodaj produkt z bazy produktów do aktualnej listy zakupów” .....	14
Tabela 2 Scenariusz przypadku użycia „Odznacz dokonany zakup produktu” .....	15
Tabela 3 Scenariusz przypadku użycia „Usuń aktualną listę zakupów” ..	16
Tabela 4 Scenariusz przypadku użycia „Dodaj produkt do bazy produktów” .....	18
Tabela 5 Scenariusz przypadku użycia „Dodaj produkt do aktualnej listy zakupów” .....	19
Tabela 6 Scenariusz przypadku użycia „Usuń produkt z aktualnej listy zakupów” .....	20
Tabela 7 Scenariusz przypadku użycia „Usuń produkt z bazy produktów” .....	21
Tabela 8 Scenariusz przypadku użycia „Wpisz nazwę miasta” .....	23
Tabela 9 Scenariusz przypadku użycia „Wpisz rodzaj jedzenia” .....	24
Tabela 10 Scenariusz przypadku użycia „Sprawdź szczegółowe informacje restauracji” .....	25
Tabela 11 Scenariusz przypadku użycia „Wyloguj się” .....	27
Tabela 12 Scenariusz przypadku użycia „Skontaktuj się poprzez odnośnik do poczty elektronicznej” .....	28
Tabela 13 Scenariusz przypadku użycia „Skontaktuj się poprzez odnośnik do konta Github” .....	29



## Spis zdjęć

Rysunek 1 Ekran rejestracji .....	30
Rysunek 2 Ekran rejestracji z błędem .....	31
Rysunek 3 Ekran logowania .....	32
Rysunek 4 Metoda przechowywania klucza lokalnie w aplikacji .....	33
Rysunek 5 Ekran logowania z błędem.....	34
Rysunek 6 Ekran aktualnej listy zakupów – pusta lista .....	35
Rysunek 7 Ekran usuwania całej aktualnej listy zakupów .....	37
Rysunek 8 Ekran bazy produktów – pusta lista .....	38
Rysunek 9 Ekran dodawania produktu do bazy .....	40
Rysunek 10 Ekran dodawania produktu do bazy – przykładowo uzupełniona nazwa produktu .....	41
Rysunek 11 Ekran bazy produktów – przykładowo uzupełniona lista .....	42
Rysunek 12 Ekran aktualnej listy zakupów – przykładowo uzupełniona lista.....	44
Rysunek 13 Ekran wyszukiwania restauracji – bez uzupełnionych pól wyszukiwania .....	46
Rysunek 14 Ekran wyszukiwania restauracji – wypełnione przykładowo pola .....	48
Rysunek 15 Ekran wyszukiwania restauracji – przykładowa lista wyników .....	49
Rysunek 16 Ekran wyszukiwania restauracji – przykładowy ekran dodatkowych informacji o wybranej restauracji .....	50
Rysunek 17 Ekran zarządzania kontem .....	51

## **Spis diagramów**

Diagram 1 Schemat bazy danych.....	10
Diagram 2 Uogólniony diagram przypadków użycia.....	12
Diagram 3 Rozwinięcie przypadku użycia „Zarządzaj aktualną listą zakupów” .....	13
Diagram 4 Rozwinięcie przypadku użycia „Zarządzaj bazą produktów” .	17
Diagram 5 Rozwinięcie przypadku użycia „Wyszukaj restauracje” .....	22
Diagram 6 Rozwinięcie przypadku użycia „Zarządzaj kontem” .....	26