

# Analiza systemu ShareShopping

Autorzy:

Michał Czerniakowski Z710

Karol Michalski Z710

Maciej Szostak Z710

## 1. Opis

Aplikacja ShareShopping będzie multi-platformową aplikacją ułatwiającą współdzielenie listy zakupów ze swoimi przyjaciółmi lub rodziną. Dzięki prostemu systemowi prześlesz swoją listę zakupów z jednego urządzenia na drugie w ciągu kilku sekund. ShareShopping będzie działać na telefonach opartych o system Android, iOS, oraz na przeglądarkach internetowych.

## 2. Krótka prezentacja systemu

Aby skorzystać z aplikacji użytkownik będzie musiał się zarejestrować przy użyciu adresu e-mail oraz hasła. Po zarejestrowaniu użytkownik będzie miał możliwość zalogowania się. Następnym krokiem będzie utworzenie pierwszej listy zakupów. Jedynym wymaganym polem będzie nazwa listy. Podczas tworzenia zostanie dla listy wygenerowane Id dzięki któremu dana lista będzie unikalna. Po pomyślnej kreacji listy będziemy mogli dodawać i usuwać produkty oraz udostępniać jej łącze lub kod QR innym użytkownikom aplikacji.

Gdy inny użytkownik otrzyma łącze do listy wyskoczy mu okno dialogowe z pytaniem czy na pewno chce dodać daną listę do swojego konta. Po potwierdzeniu będzie widział produkty dodane przez innych użytkowników oraz będzie mógł dodawać i usuwać produkty. Po każdej zmianie na liście będzie ona zsynchronizowana u wszystkich użytkowników mających do niej dostęp, dzięki czemu każdy będzie widział miał te same dane.

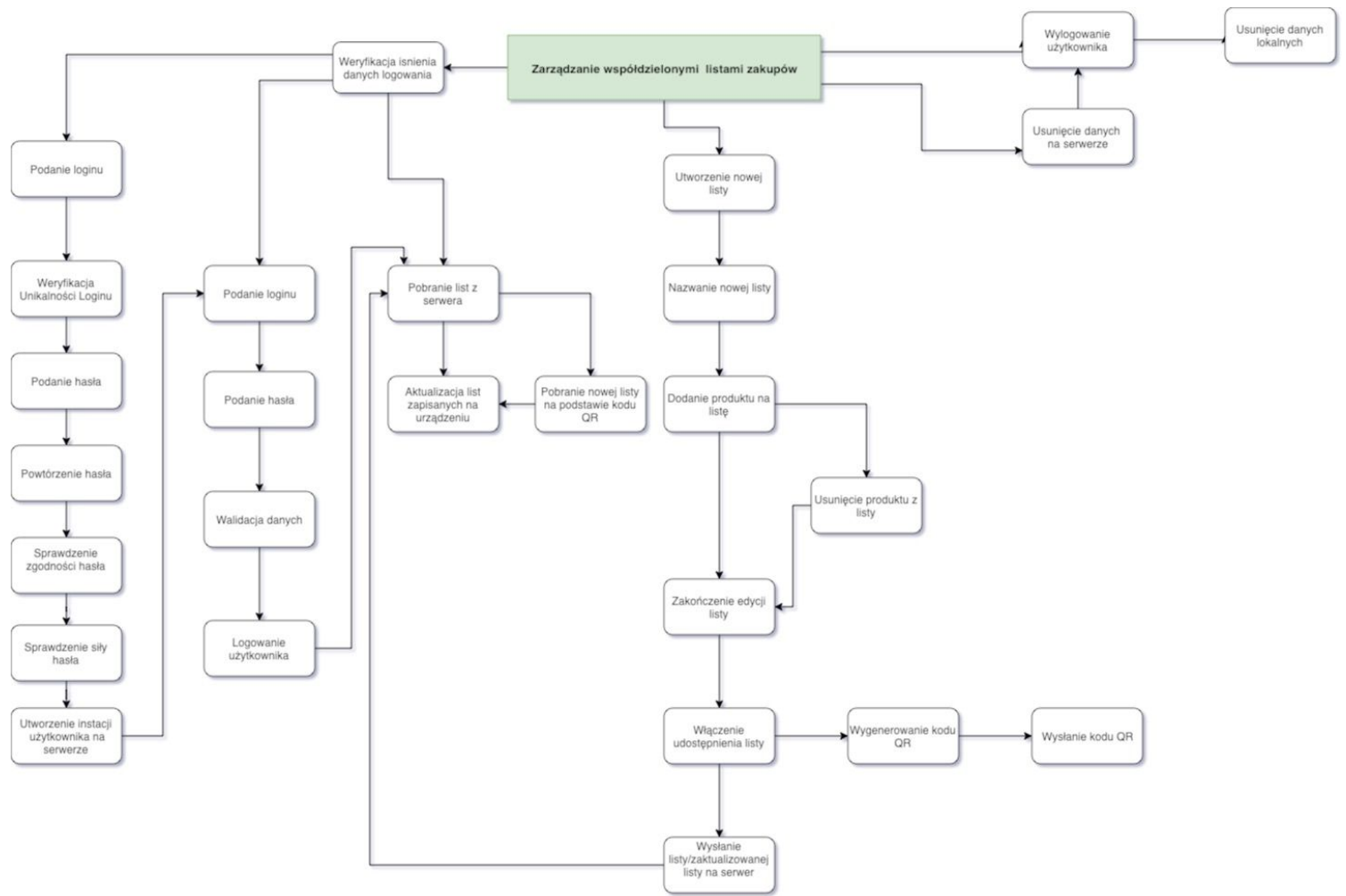
Kod QR ma jeszcze bardziej upraszczać dzielenie się listą zakupów. Wystarczy że użytkownik chcący udostępnić swoją listę wyświetli jej kod QR na ekranie swojego telefonu, a drugi użytkownik zeskanuje kod przy pomocy wbudowanego skanera QR.

# Specyfikacja funkcjonalna

## **1. Lista funkcji z opisem ich zastosowania**

- 1.1. Weryfikacja istnienia danych logowania
- 1.2. Logowanie użytkownika
  - 1.2.1. Podanie loginu
  - 1.2.2. Podanie hasła
  - 1.2.3. Walidacja danych
- 1.3. Rejestracja nowego użytkownika
  - 1.3.1. Podanie loginu
  - 1.3.2. Weryfikacja unikalności loginu
  - 1.3.3. Podanie hasła
  - 1.3.4. Powtórzenie hasła
  - 1.3.5. Sprawdzenie zgodności hasła
  - 1.3.6. Sprawdzanie siły hasła
  - 1.3.7. Utworzenie użytkownika na serwerze
- 1.4. Pobranie list z serwera
- 1.5. Pobranie udostępnionej listy na podstawie kodu QR
- 1.6. Aktualizacja list zapisanych na urządzeniu
- 1.7. Utworzenie listy lokalnej
  - 1.7.1. Nazwanie listy
  - 1.7.2. Dodanie produktu na listę
  - 1.7.3. Usunięcie produktu z listy
  - 1.7.4. Zakończenie edycji listy
  - 1.7.5. Umożliwienie udostępnienia listy
  - 1.7.6. Wygenerowanie kodu QR
  - 1.7.7. Udostępnienie kodu QR
- 1.8. Wysłanie zaktualizowanych list na serwer
- 1.9. Wylogowanie użytkownika
  - 1.9.1. Usunięcie danych lokalnych
- 1.10. Usunięcie konta
  - 1.10.1. Usunięcie danych lokalnych oraz na serwerze

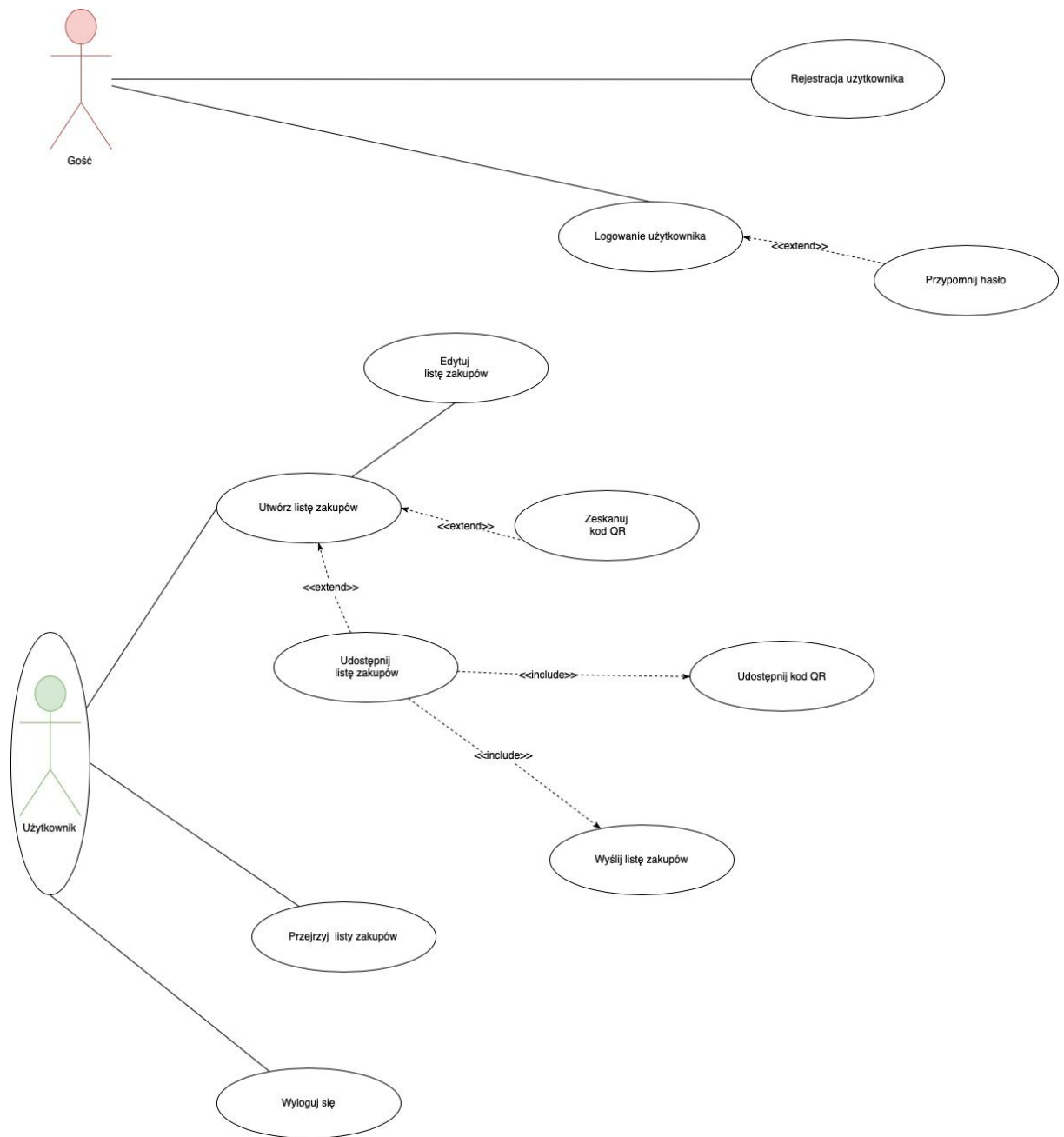
## 2. Diagram Hierarchii Funkcji (FHD)



### 3. Tabela Aktorów do Diagramu Przypadków Użycia

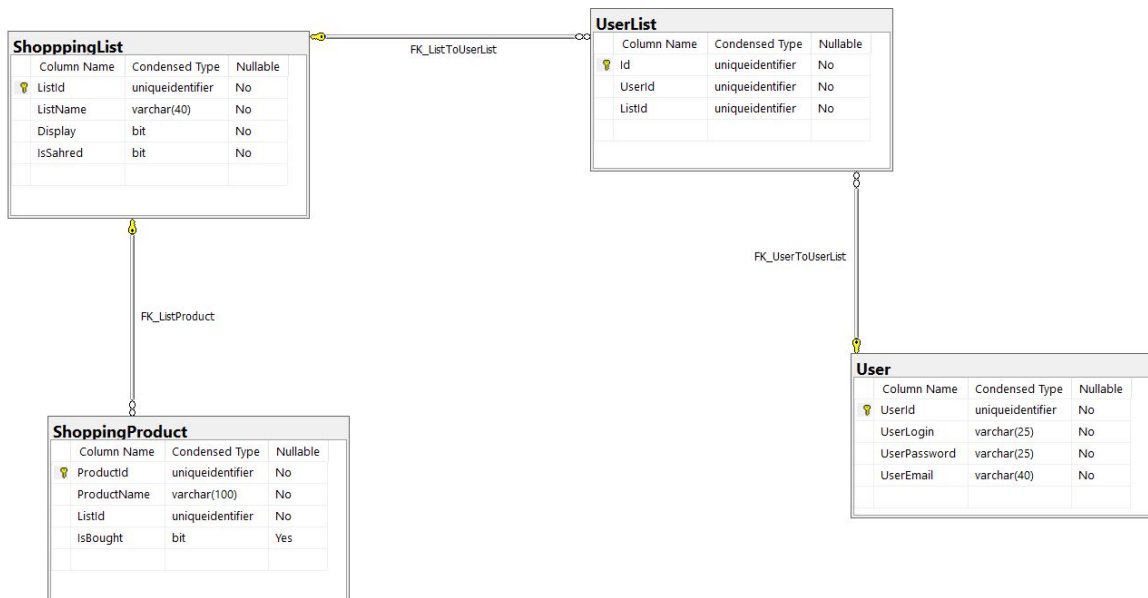
Aktor	Opis
Gość	Osoba nieposiadająca konta lub nie zalogowana
Użytkownik	Osoba posiadająca konto zalogowana do systemu

#### 4. Diagram przypadków użycia



# Model danych

## 1. Diagram ERD



## 2. Opis encji

### Tabela ShoppingList

Tabela zawierająca informacje o liście jej nazwie oraz tym czy lista jest udostępniana innym użytkownikom

### Tabela UserList

Tabela pośrednia łącząca użytkownika z listą zakupów

### Tabela User

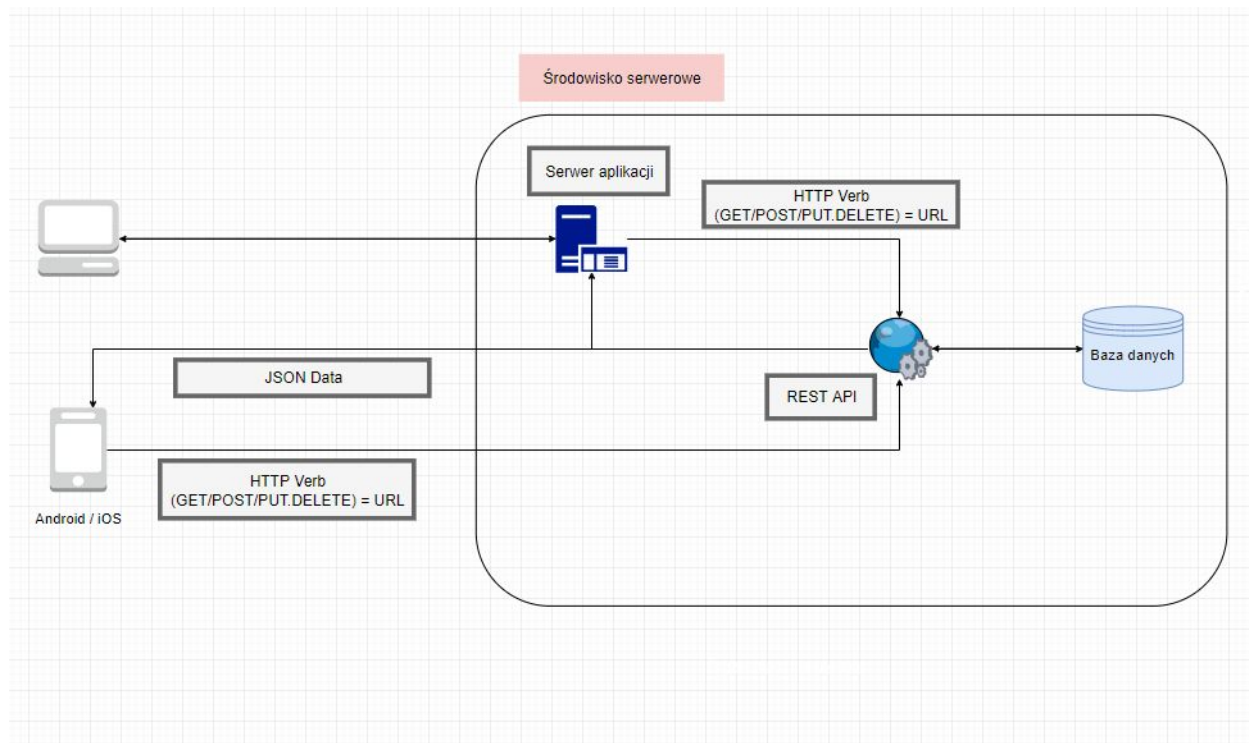
Tabela zawierająca informacje o użytkowniku login, hasło,e-mail

### Tabela ShoppingProduct

Tabela zawierająca informacje o produktach na liście produktów i statusie czy produkt jest już zakupiony



# Model architektury systemu



Architektura systemu opiera się na REST API które operuje na danych jest to o tyle wygodne że dla aplikacji mobilnej i desktopowej istnieje jeden interfejs do bazy danych przez który wszyscy klienci się łączą. Jest to rozwiązanie skalowalne bo w momencie kiedy przyjdzie dodatkowy klient aplikacji np. WPF, WinForms etc. dołączany jest interfejs API do tego klienta i de facto powstaje kolejne rozwiązanie na innej platformie. Aplikacja mobilna łączy się bezpośrednio do API restowego natomiast klienci aplikacji web będą się łączyć do serwera aplikacji na którym będzie stała strona web a serwer ten poprzez wywołanie API będzie pobierał dane i je “konsumował” wyświetlając je po stronie klienta.