

Pan Bajgiel

Aplikacja lojalnościowa dla klientów punktów gastronomicznych

“Pan Bajgiel”



Sylwia Zoń

Anna Wójcik

Karolina Pieszczek

Agata Kawalec

Spis Treści

Lista zebranych wymagań	4
Opis systemu	4
Funkcjonalności aplikacji dla klienta	4
Funkcjonalności aplikacji dla sprzedawcy	4
Opis promocji	4
Instalacja oraz użytkowanie	5
Backend	5
Testy	6
Frontend	7
Architektura	8
Baza danych	9
Serwis API	10
Model REST	10
Frameworki	11
Aplikacje mobilne	11
Frameworki	11
Wykorzystane biblioteki oraz komponenty	11
Opisy algorytmów	12
Logowanie	12
Rejestracja	13
Zakup	14
Odbiór nagrody	15
Diagramy sekwencji	16
Logowanie	16
Rejestracja	17
Odbiór nagrody	18
Aplikacja dla sprzedawcy	19
Aplikacja dla klienta	25
Lista wykorzystanych technologii	38
React Native	38
Spring	38
MySQL	38
Projekt testów	39
Testy manualne	39
Testy automatyczne	39

Wykorzystane narzędzia	40
Analiza ryzyka	41
Lista ryzyk	41
Macierz ryzyka	42
Sposoby reagowania na ryzyko	42
Lista narzędzi użytych przy realizacji projektu	44

Lista zebranych wymagań

Opis systemu

Projekt ma na celu stworzenie aplikacji lojalnościowej dla klientów punktów gastronomicznych "Pan Bajgiel". Jego zadaniem jest ułatwienie i przyspieszenie procesu zakupu produktów oraz optymalizacji sprzedaży poprzez wprowadzenie aplikacji mobilnych dla klientów oraz sprzedawców.

Funkcjonalności aplikacji dla klienta

- indywidualny profil z kartą lojalnościową w postaci kodu QR, umożliwiającą zbieranie punktów podczas zakupów
- graficzny spis oferowanych produktów wraz z aktualną ceną
- mapa z naniesionymi punktami sprzedaży
- możliwość aktywowania promocji po zebraniu określonej ilości punktów
- możliwość zmiany hasła
- zakładanie oraz usuwanie konta

Funkcjonalności aplikacji dla sprzedawcy

- dostęp do statystyk sprzedaży we wszystkich punktach sprzedaży, co pozwoli na lepsze zarządzanie biznesem (optymalizacja zawartości dostaw, dostępnych promocji itd.)
- możliwość skanowania karty lojalnościowej klienta wbudowanym w urządzenie mobilne aparatem przy każdych zakupach
- możliwość wprowadzania rodzaju zakupów po zeskanowaniu karty lojalnościowej
- możliwość zmiany hasła

Opis promocji

- gratis - darmowy bajgiel - za sfinalizowanie dziesięciu kolejnych zakupów z wykorzystaniem aplikacji. Za każdy zakup klient otrzyma jedną pieczętkę w aplikacji, pieczętki dodawane są tylko w przypadku braku oczekującego do odbioru gratisu.
- zniżka procentowa - aktywacja zniżki w zamian za zebrane punkty (1pln=1pkt), 10pkt = 1% zniżki, 20pkt = 2% zniżki etc. po odebraniu rabatu punkty są odejmowane z puli posiadanej przez użytkownika.

Instalacja oraz użytkowanie

Backend

Backend dostarczony jest formie aplikacji napisanej w Java Spring Boot. Dla użytkownika dostępny jest jako serwis restowy pod adresem ip: 104.211.12.200 oraz portem: 8081.

Adresy endpoint'ów:

- Produkty:
 - metoda get:
 - <http://104.211.12.200:8081/product>
 - <http://104.211.12.200:8081/product?id=1> - przykładowy id produktu
 - metoda post:
 - <http://104.211.12.200:8081/transaction>
 - Body:

```
{
  "userLogin": "user@mail.com",
  "shopId": "1",
  "date": "2020-01-20"
}
```
 - <http://104.211.12.200:8081/transactionDetails>
 - Body:

```
{
  [
    {
      "transactionId": "1",
      "productId": "1",
      "amount": "2"
    }
  ]
}
```
- Sklepy:
 - metoda get:
 - <http://104.211.12.200:8081/shop>
 - <http://104.211.12.200:8081/shop?id=1>
- Statystyki:
 - metoda get:
 - <http://104.211.12.200:8081/shops>
 - <http://104.211.12.200:8081/baigielis>
 - <http://104.211.12.200:8081/shop>

- Użytkownik:
 - metoda get:
 - <http://104.211.12.200:8081/user?login=user@mail.com>
 - <http://104.211.12.200:8081/user/login?login=user@mail.com&password=pass>
 - <http://104.211.12.200:8081/user/update/points?login=user@mail.com&points=10>
 - <http://104.211.12.200:8081/user/update/stamps?login=user@mail.com&stamps=10>
 - metoda post:
 - <http://104.211.12.200:8081/user/password>
 - Body:


```
{
                "userLogin": "user@mail.com",
                "password": "pass",
              }
```
 - <http://104.211.12.200:8081/user/register>
 - Body:


```
{
                "userLogin": "user@mail.com",
                "password": "pass",
                "name": "User",
              }
```
 - metoda delete:
 - <http://104.211.12.200:8081/user?login=user@mail.com>

Aby uruchomić aplikację na lokalnym urządzeniu należy uruchomić metodę main z klasy Main.

Testy

Testy integracyjne oraz jednostkowe kontrolerów dostępne są w pakiecie tests.

Docker

Na wirtualnej maszynie Azure działają dwa kontenery Dockera. W jednym z nich znajduje się baza danych, w drugim aplikacja serwisu.

Plik docker-compose:

```
version: '3'
services:

  web:
    build: .
    ports:
      - 8081:8081

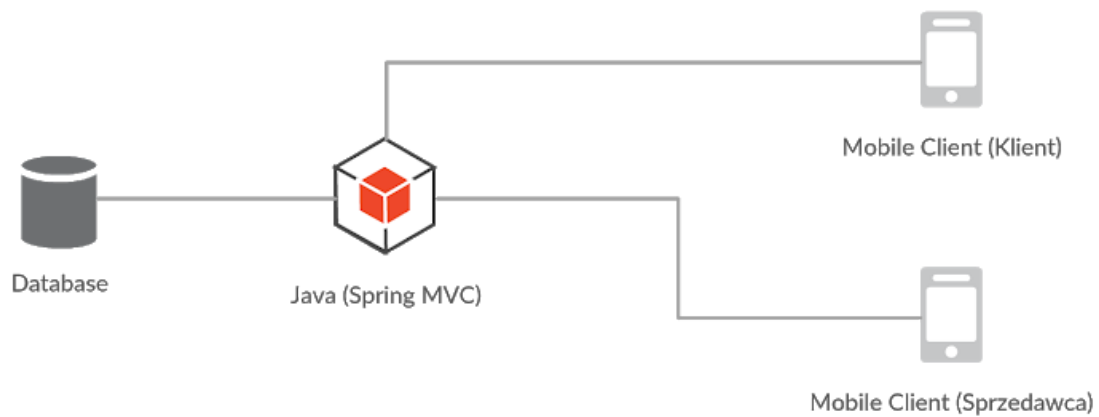
  db:
    image: mysql
    ports:
      - 3306:3306
    volumes:
      - /var/panbajgiel/data:/var/lib/mysql
    environment:
      - MYSQL_DATABASE=panbajgiel
      - MYSQL_USER=sylwia
      - MYSQL_PASSWORD=strongDBP4ssword
      - MYSQL_ROOT_PASSWORD=strongDBP4ssword
      - MYSQL_ROOT_HOST=%
```

Frontend

Użytkownikom - klientom dostarczona jest aplikacja w formacie apk. Aplikację należy zainstalować na urządzeniu mobilnym.

Użytkownicy aplikacji dla sprzedawcy nie zajmują się instalacją. Zainstalowana aplikacja dostarczana jest wraz z urządzeniem mobilnym.

Architektura

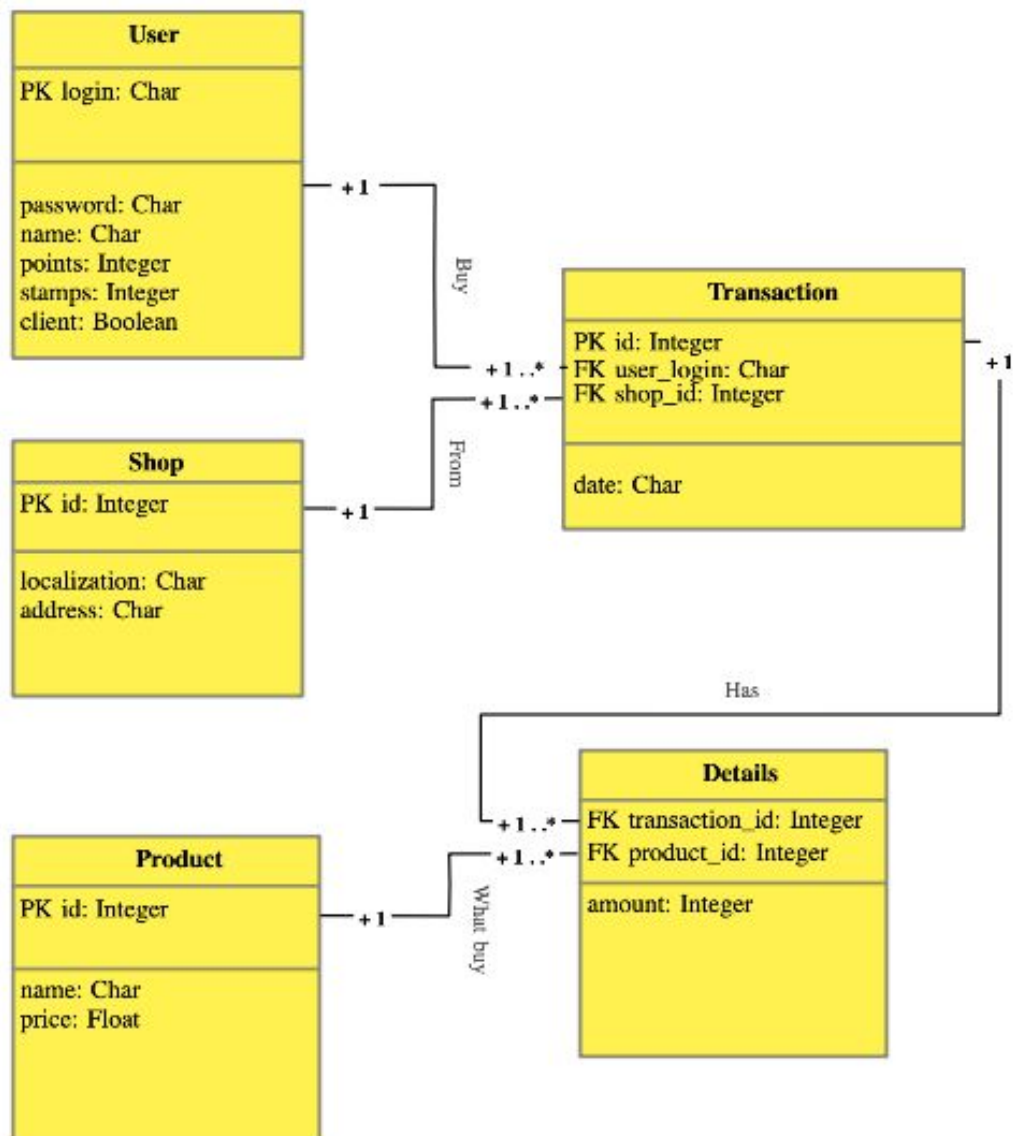


W skład naszego oprogramowania wchodzić będzie baza danych, backend oraz dwie aplikacje mobilne.

Baza danych

Baza danych zawiera dane o klientach oraz sprzedawcach, są potrzebne do weryfikacji logowania. Tabela *Shop* przechowuje dane dotyczące lokalizacji naszych sklepów, a tabela *Product* magazynuje informacje o rodzajach produktów, które oferujemy.

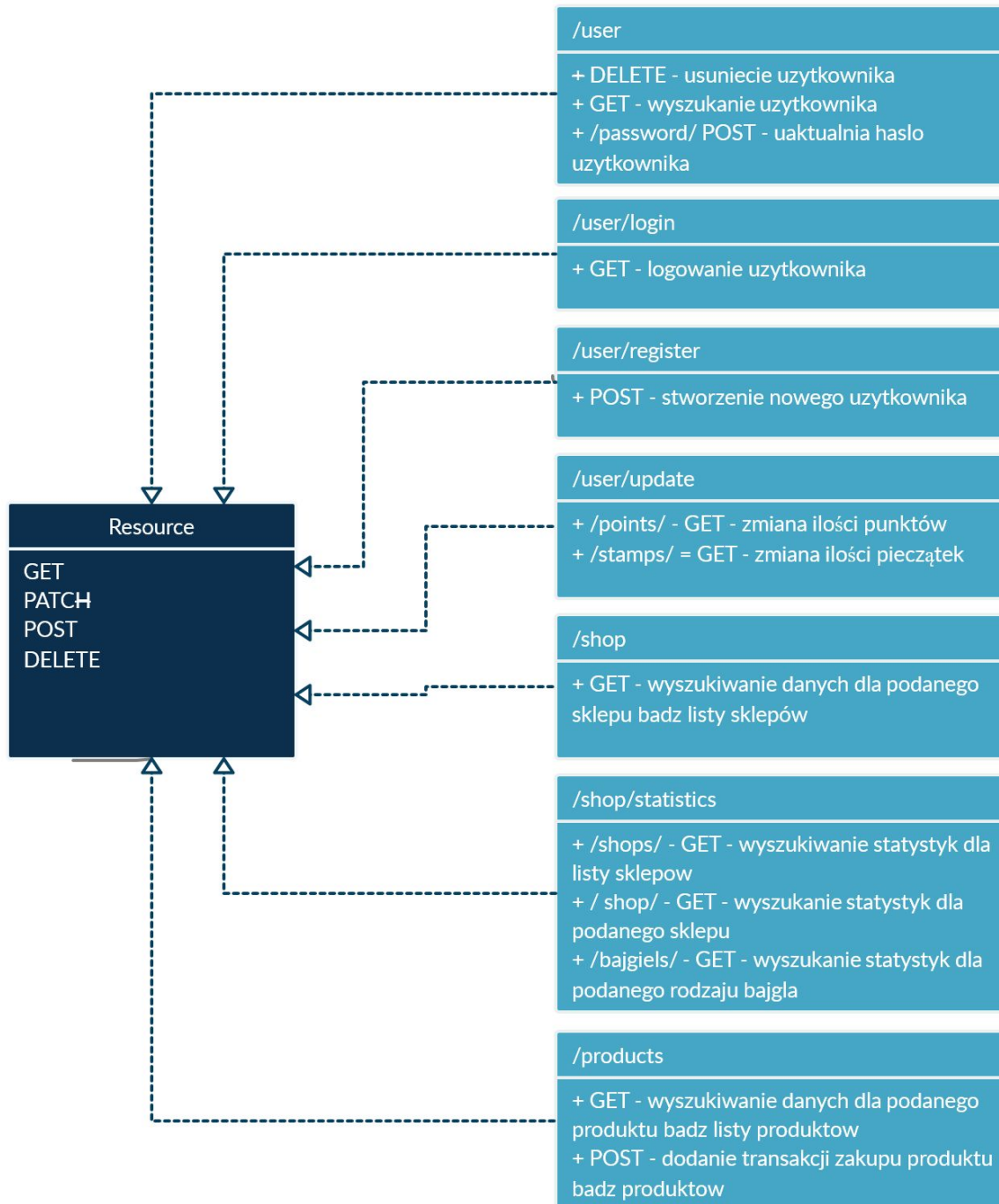
Transaction oraz *Details* to najważniejsze tabele, dzięki którym będziemy mogli śledzić między innymi, które z naszych produktów są najczęściej kupowane, czy które stoiska są najczęściej odwiedzane.



Serwis API

Model REST

(Interfejs RESTowy serwisu backendowego)

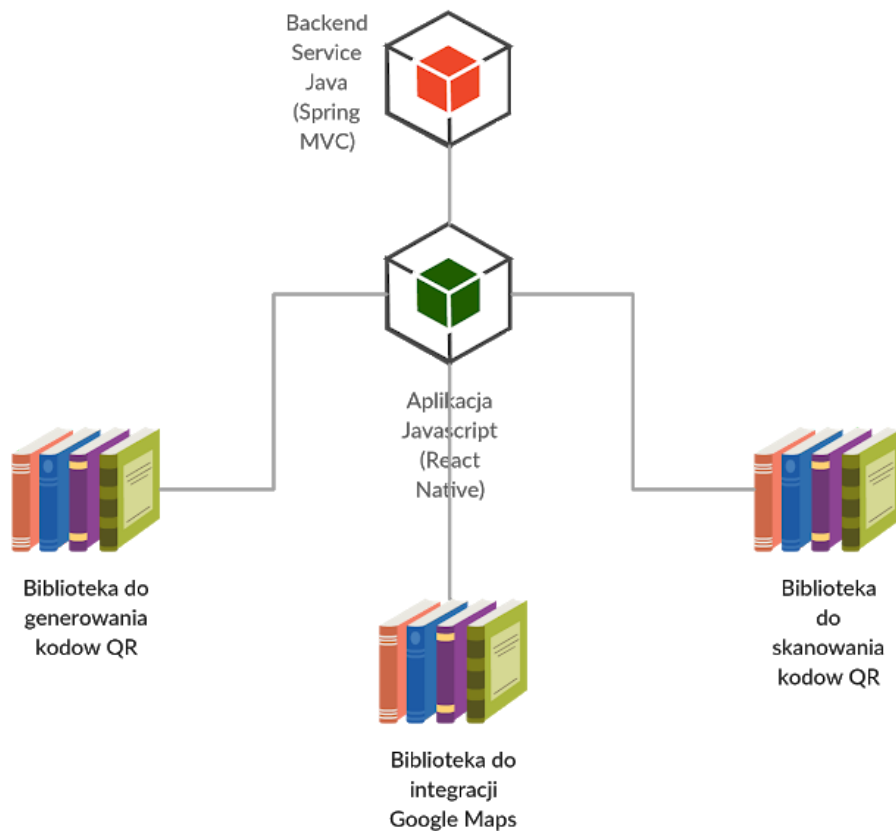


Frameworki

- Spring

Aplikacje mobilne

Aplikacja dla użytkownika - klienta oraz dla sprzedawcy.



Frameworki

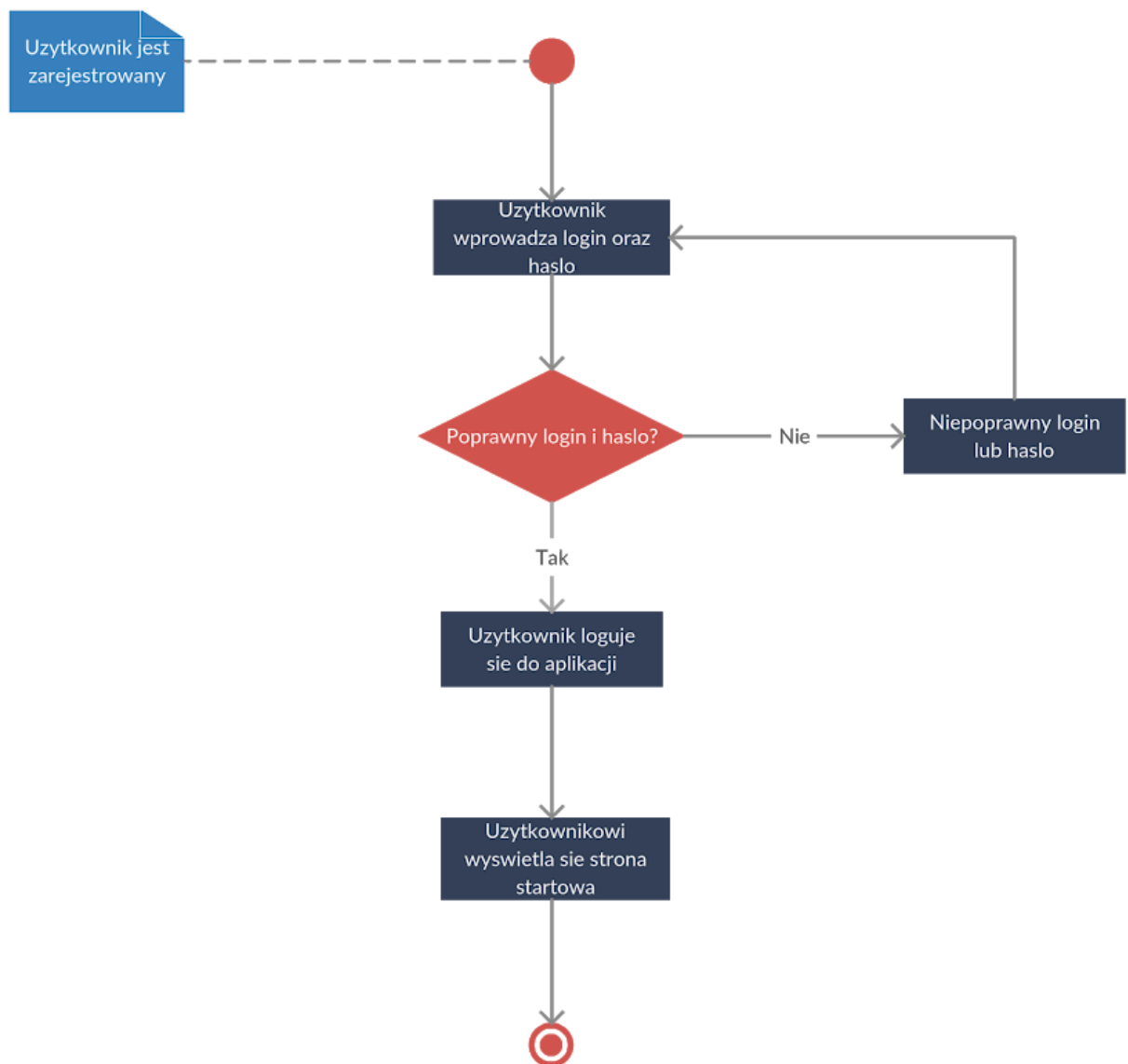
- React Native

Wykorzystane biblioteki oraz komponenty

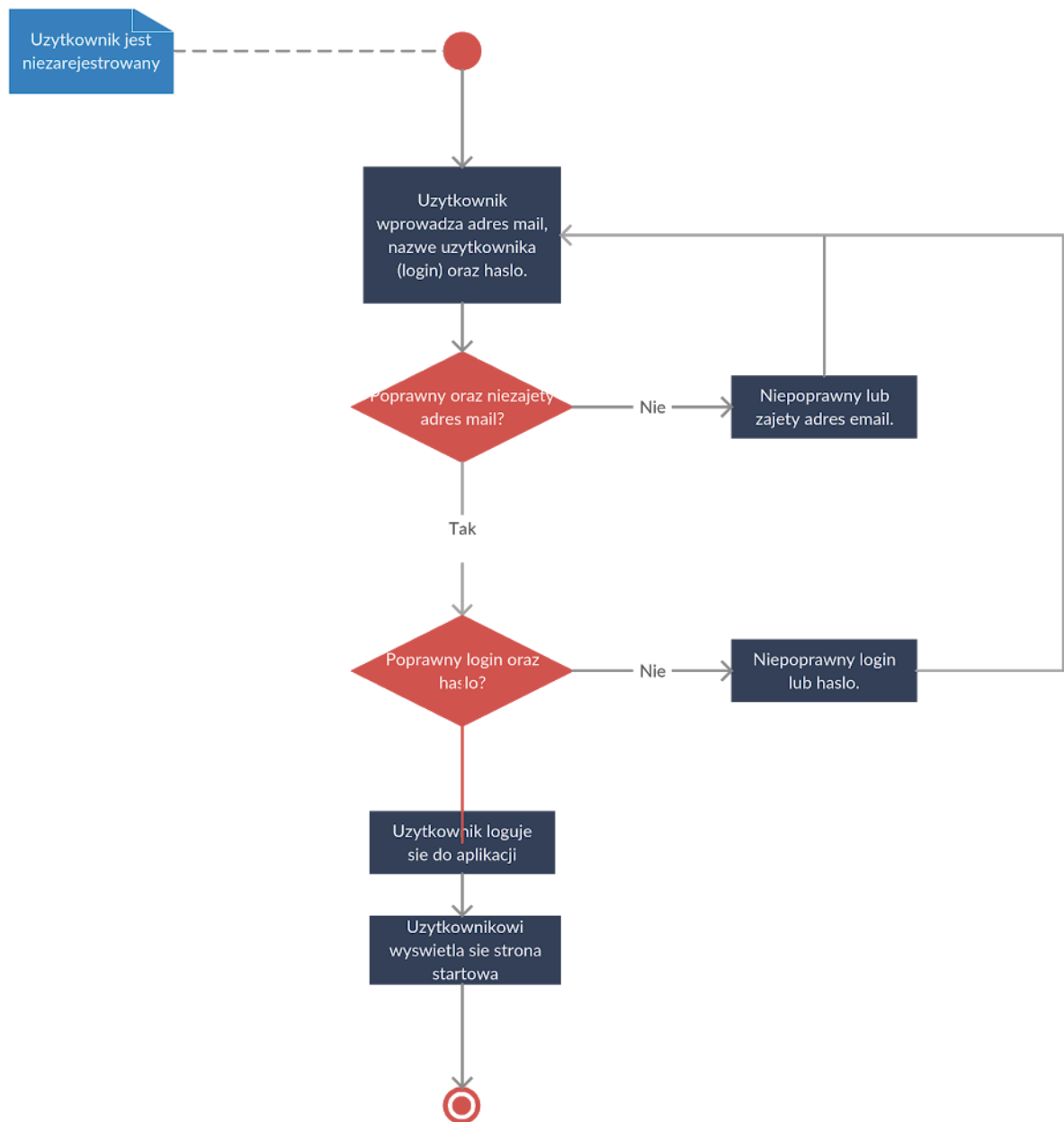
- <https://github.com/moaazsidat/react-native-qrcode-scanner>
Komponent wykorzystywany do czytania kodów QR.
- <https://github.com/react-native-community/react-native-maps>
react-native-maps - biblioteka do integracji Google Maps z aplikacją napisaną w React Native
- <https://www.npmjs.com/package/react-native-qrcode-svg>

Opisy algorytmów

Logowanie



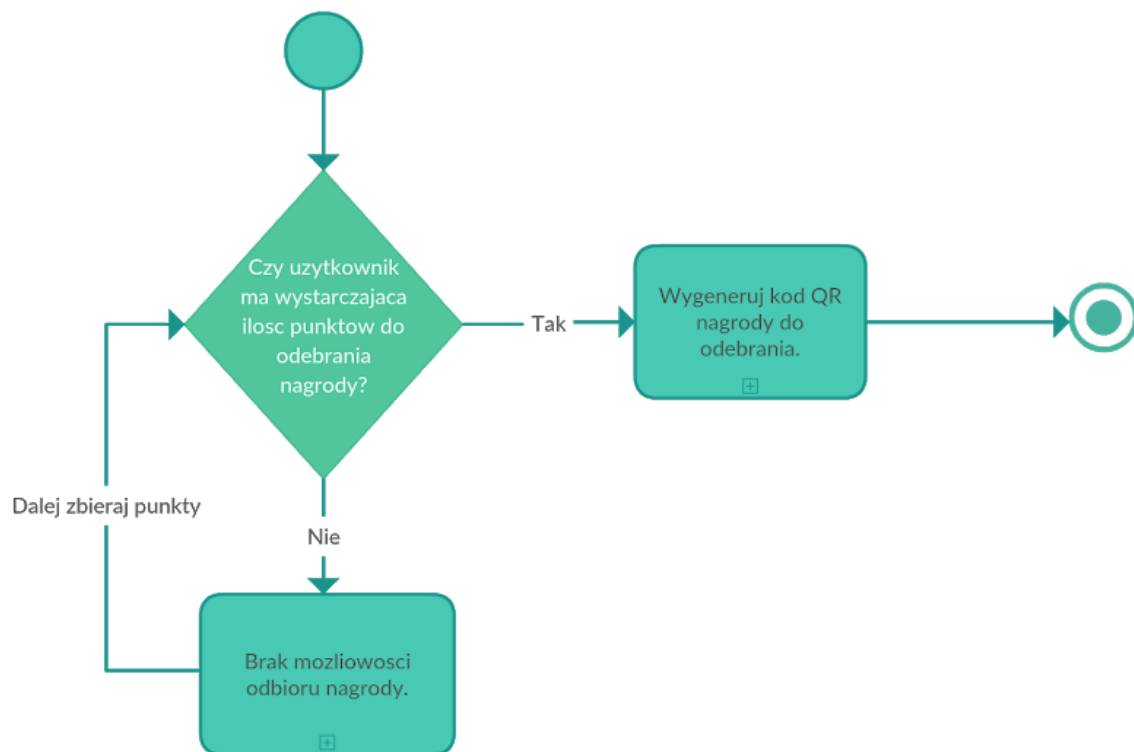
Rejestracja



Zakup

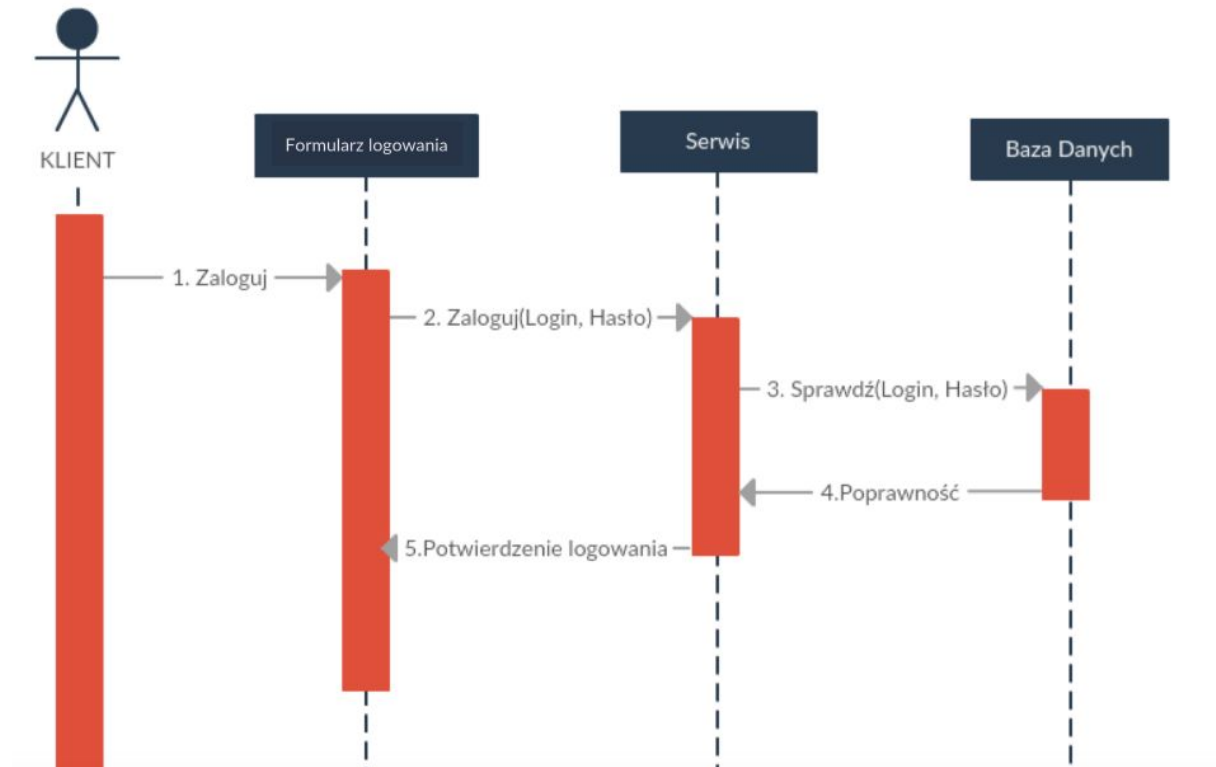


Odbiór nagrody

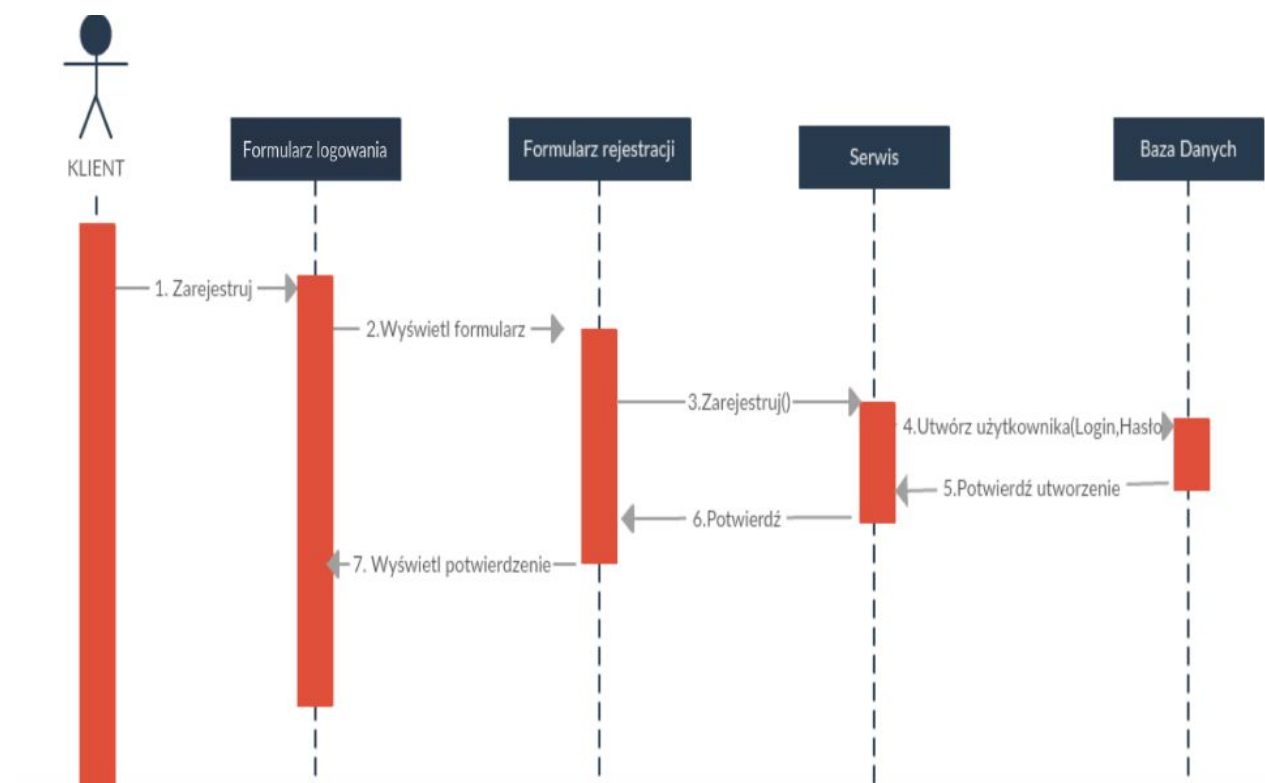


Diagramy sekwencji

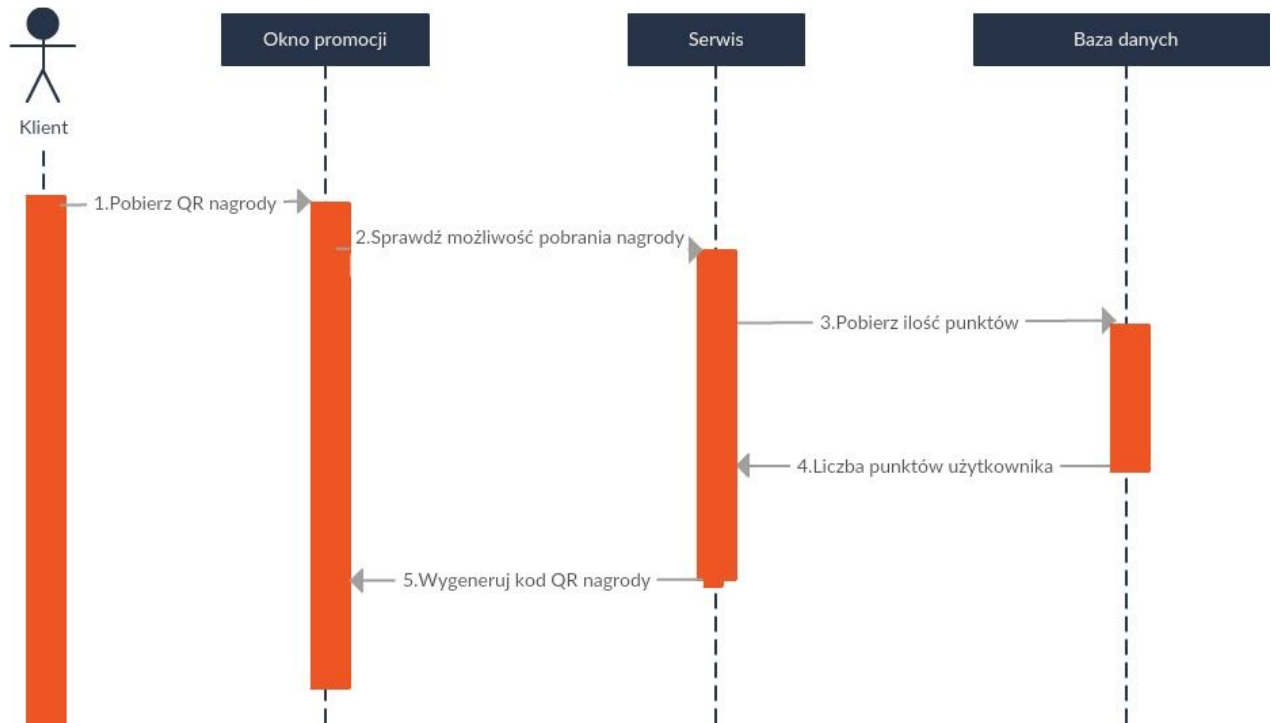
Logowanie



Rejestracja



Odbiór nagrody



Aplikacja dla sprzedawcy

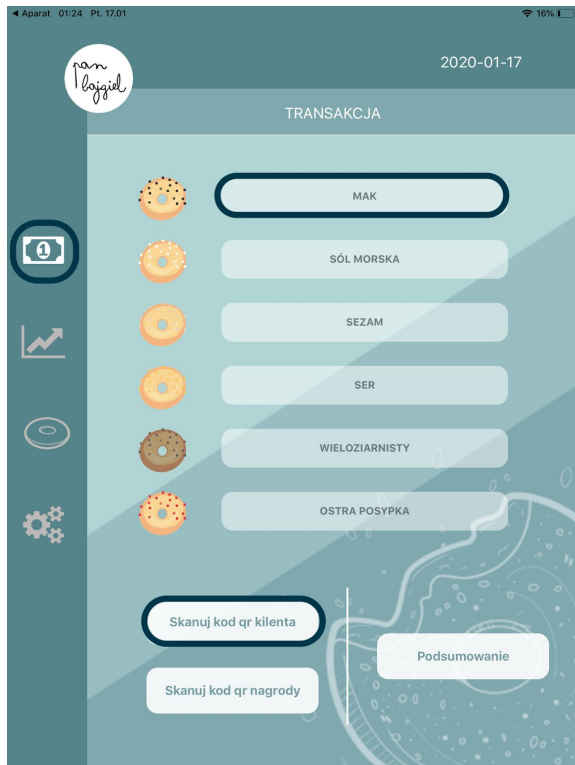


Ekran ładowania aplikacji



Ekran logowania

Sprzedawca nie zakłada konta sam. Otrzymuje potrzebne dane do zalogowania, następnie wpisuje je w polach *login* oraz *hasło*. Przycisk zaloguj przełącza nas do części właściwej aplikacji.

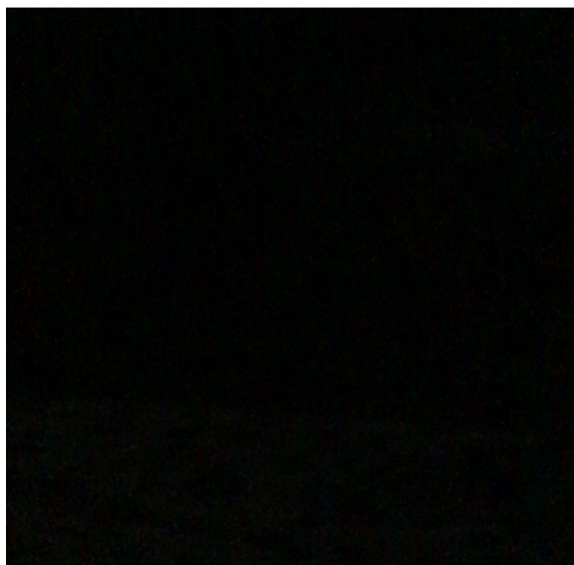


Transakcja

Dokonanie zakupu. Należy w odpowiednich polach wpisać liczbę zakupionych produktów. W przypadku gdy zakup nie obejmuje danego rodzaju produktu, pole zostawiamy puste.

Następnie przechodzimy do skanowania kodu QR klienta i opcjonalnie kodu QR nagrody klienta.

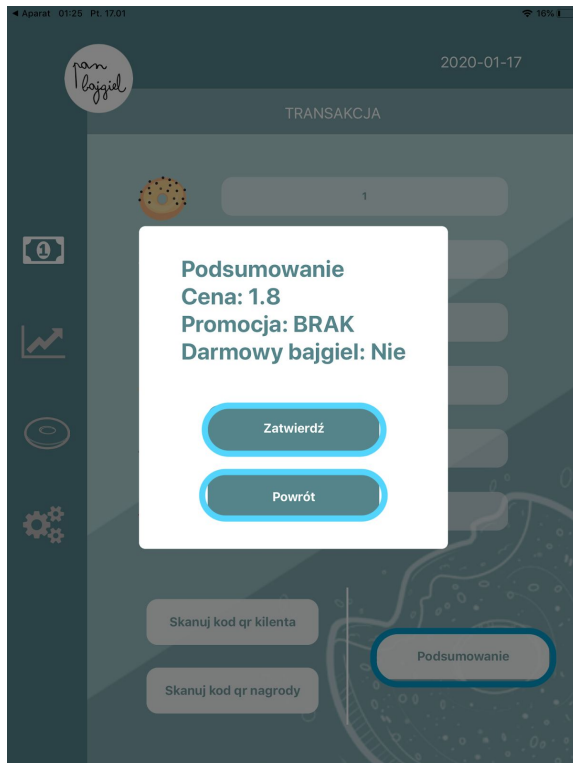
Zeskanuj kod QR użytkownika



Skanowanie kodu QR użytkownika

Po nakierowaniu kamery na kod QR pojawia się potwierdzenie zeskanowania. Następnie przyciskiem **powrót** wracamy do dokończenia transakcji.

W ten sam sposób sczytujemy kod QR nagrody.

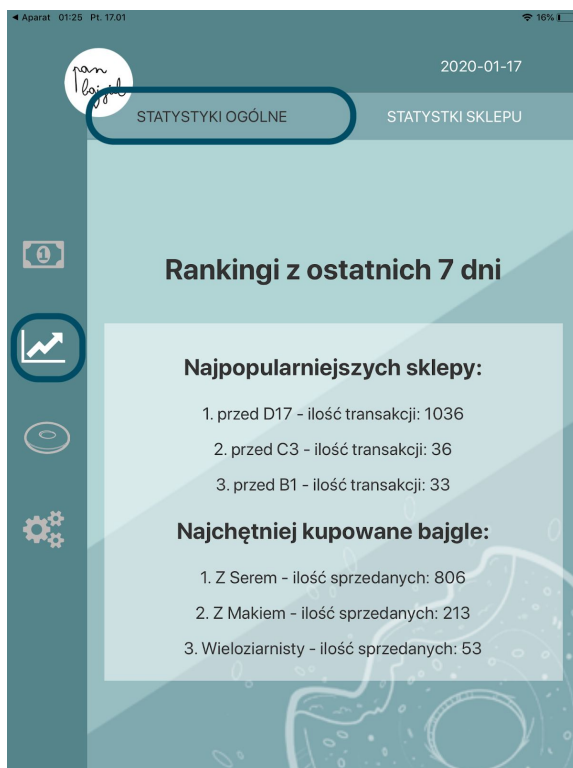


Podsumowanie

Po zeskanowaniu kodów oraz wpisaniu ilości zakupionych produktów, wciskamy **podsumowanie**. W wyskakującym okienku pojawia nam się:

- cena za cały zakup z uwzględnieniem promocji
- informacja o wielkości promocji lub BRAK w przypadku jej braku
- informacja o darmowym produkcie

Aby sfinalizować transakcję wciskamy **zatwierdź**. W celu poprawienia lub zmiany danych wciskamy **powrót**.





Statystyki

Statystyki sklepu. Informacja o wielkości sprzedaży z ostatnich 7 dni. Takie statystyki zrobione są dla każdego sklepu.





Cennik

Informacja o cenie produktów w ofercie.



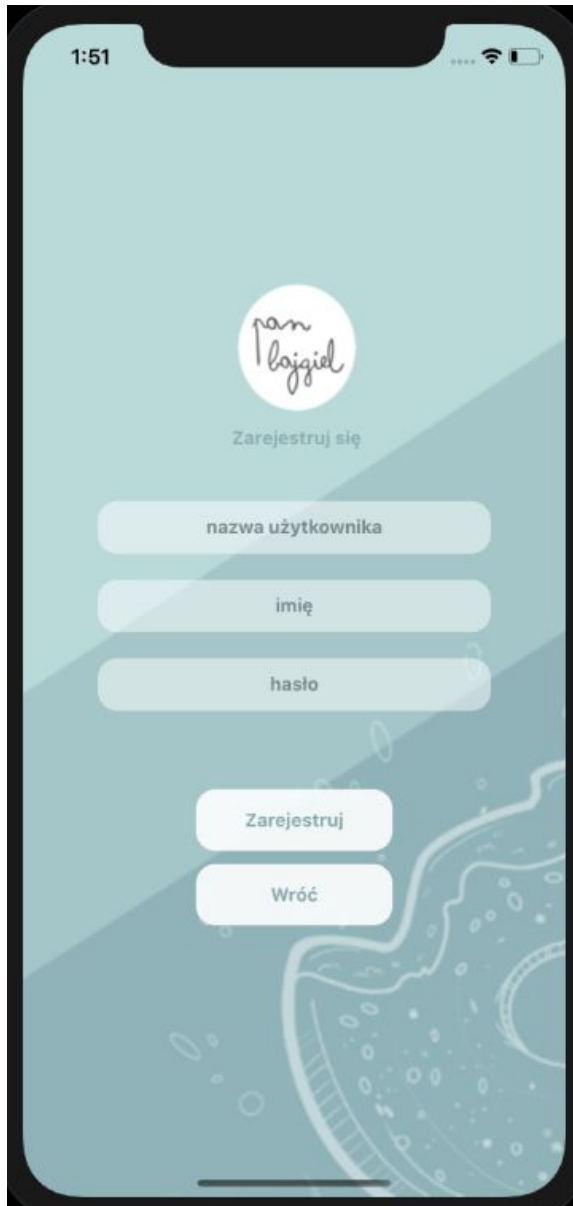
Ustawienia

Przycisk **wyloguj się** wylogowuje obecnego użytkownika.

Przycisk **zmień hasło** otwiera formularz zmiany hasła. Należy podać dwukrotnie nowe hasło i zaakceptować.



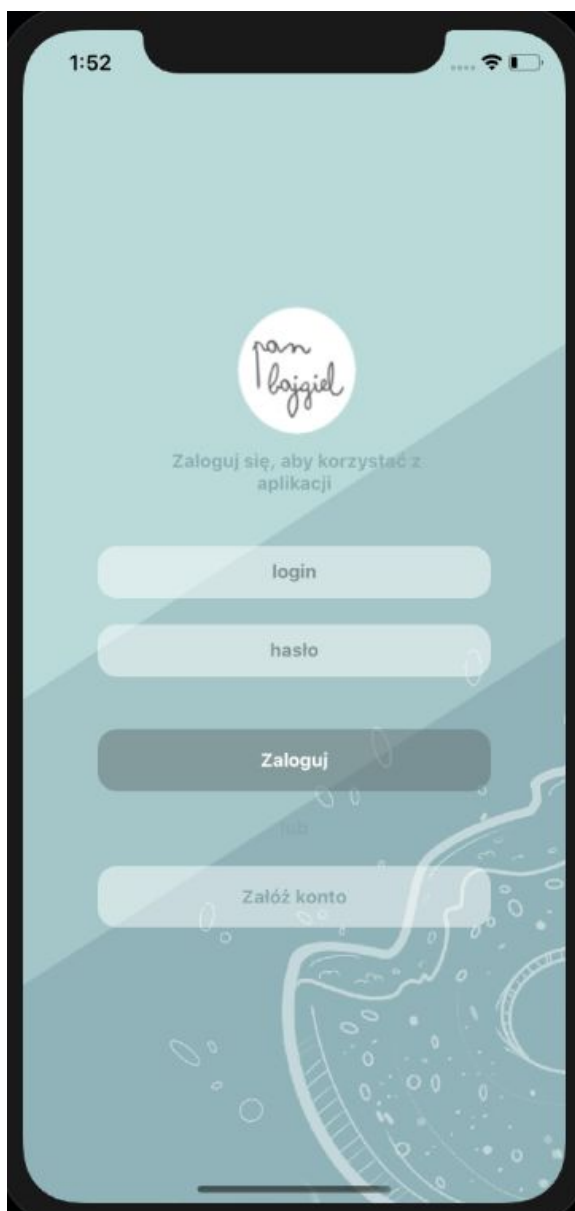
Aplikacja dla klienta



Rejestracja

Jeśli jeszcze nie masz konta w naszej aplikacji, tak właśnie wygląda formularz rejestracyjny, który będziesz musiał wypełnić.

W polu *nazwa użytkownika* należy podać mail, w polu *imię* - Twoje imię, oraz *hasło*. Aby wejść w formularz rejestracyjny należy wybrać przycisk **Założ konto** z widoku logowania, który jest opisany poniżej



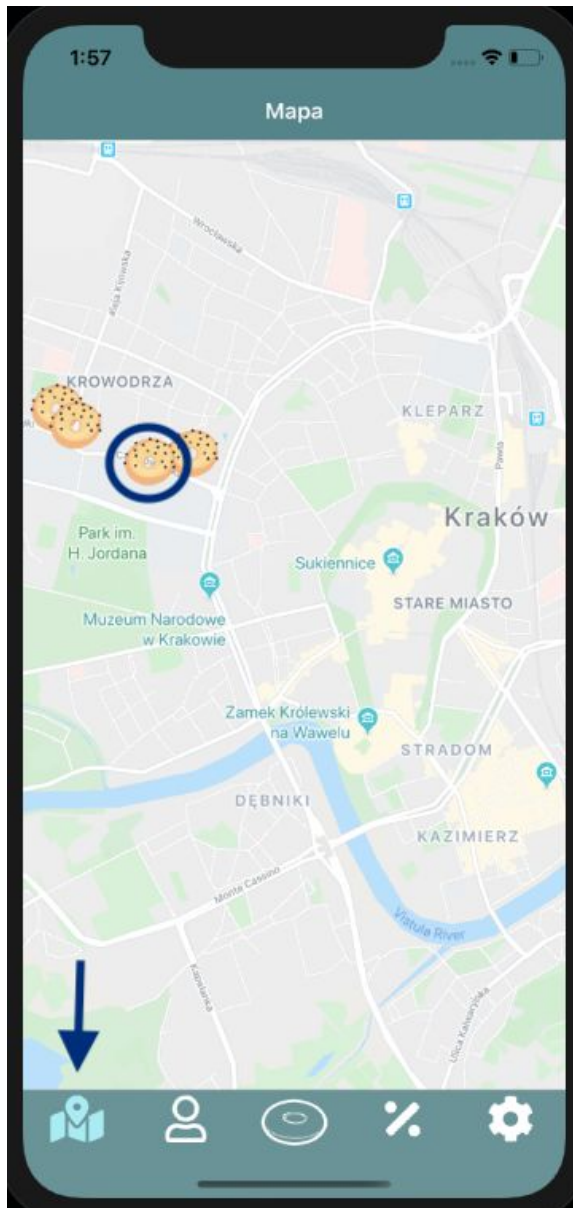
Logowanie

W celu zalogowania się do aplikacji, należy podać *login* (mail podany przy rejestracji) oraz *hasło*, a następnie kliknąć **Zaloguj**



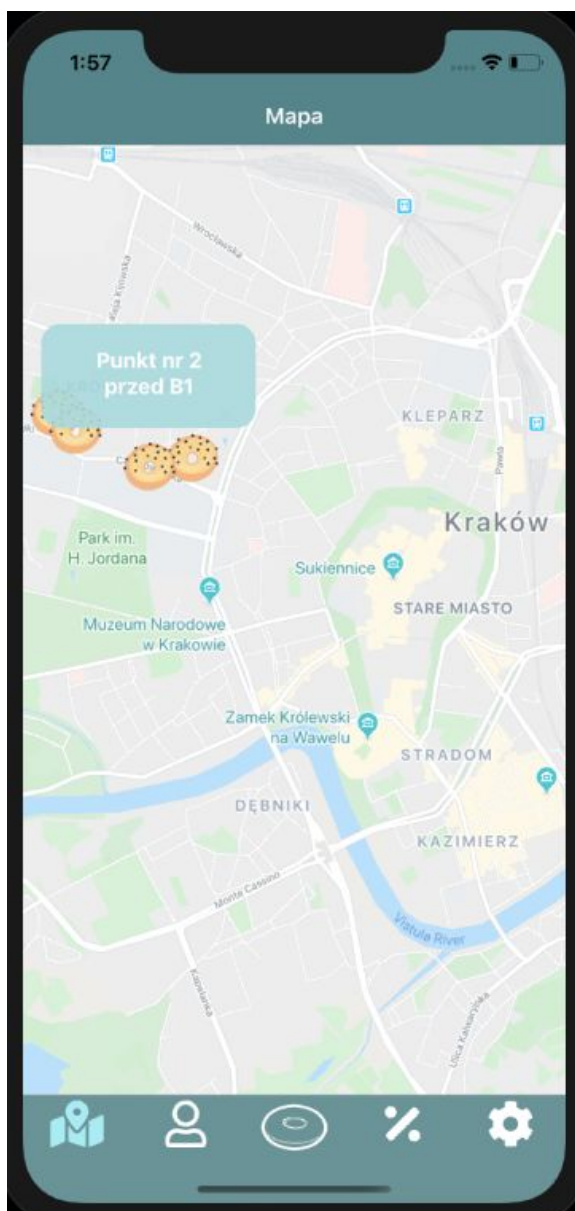
Logowanie

W przypadku podania błędnego hasła, zostaniesz o tym poinformowany poprzez stosowny komunikat



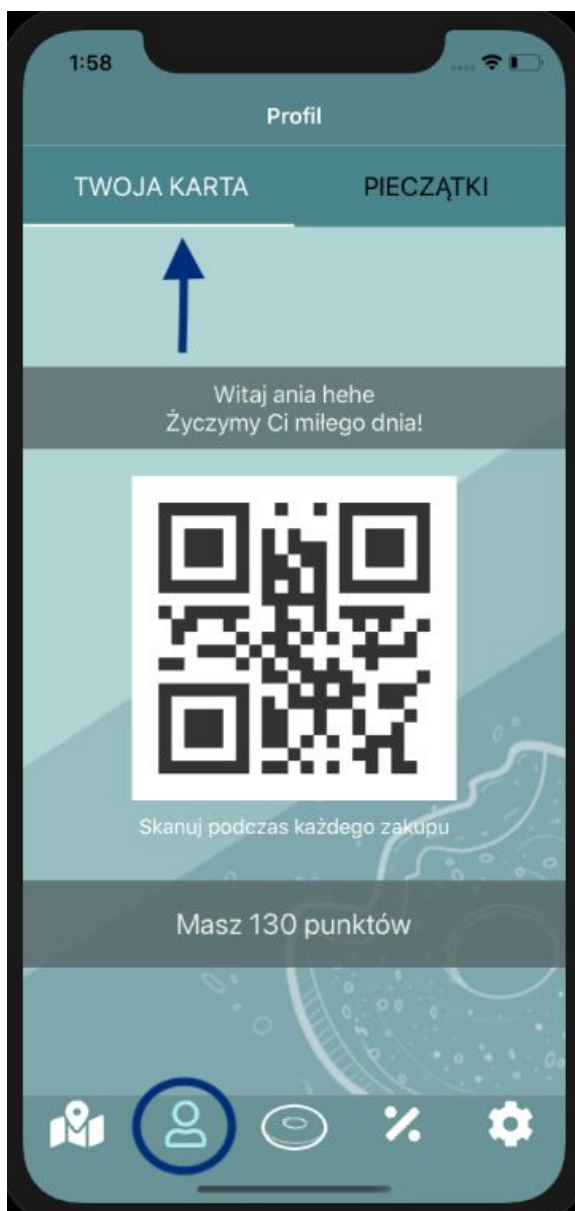
Mapy

Po prawidłowym zalogowaniu zostaniesz przekierowany do aplikacji głównej. Pierwszym ekranem są Mapy. Na dole ekranu znajdziesz [menu nawigacyjne](#). Ikona wskazana na rysunku strzałką odpowiada bieżącemu widokowi. [Ikony bajgli](#) na mapie - tutaj zaznaczone kółkiem - odpowiadają naszym punktom sprzedaży.



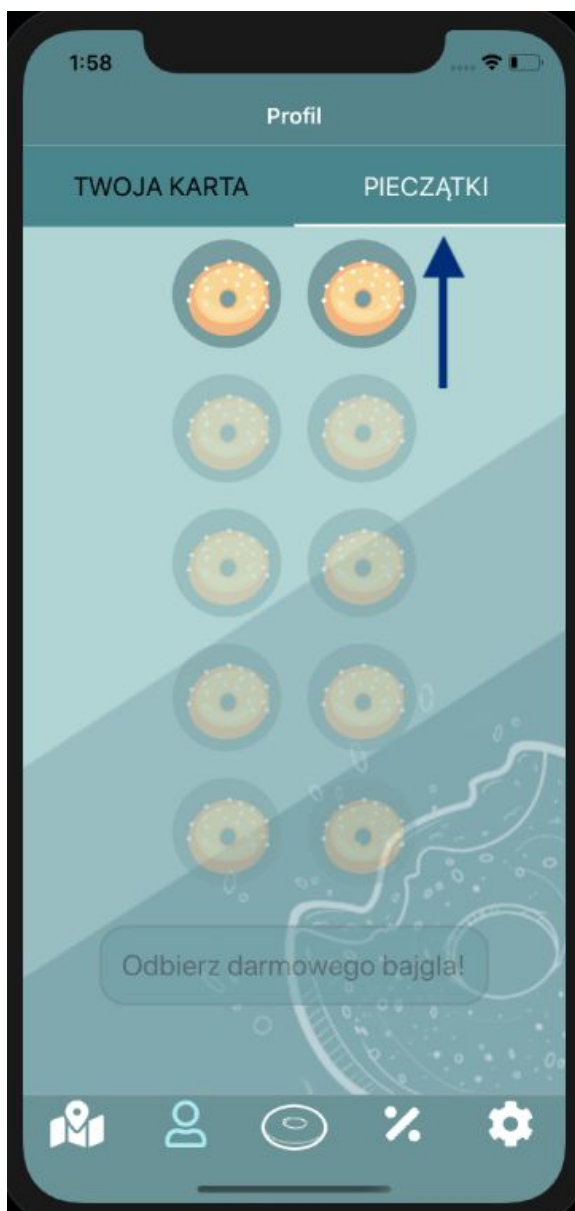
Mapy

Klikając na *bajgla* na mapie, otrzymujesz informację o wybranym punkcie sprzedaży



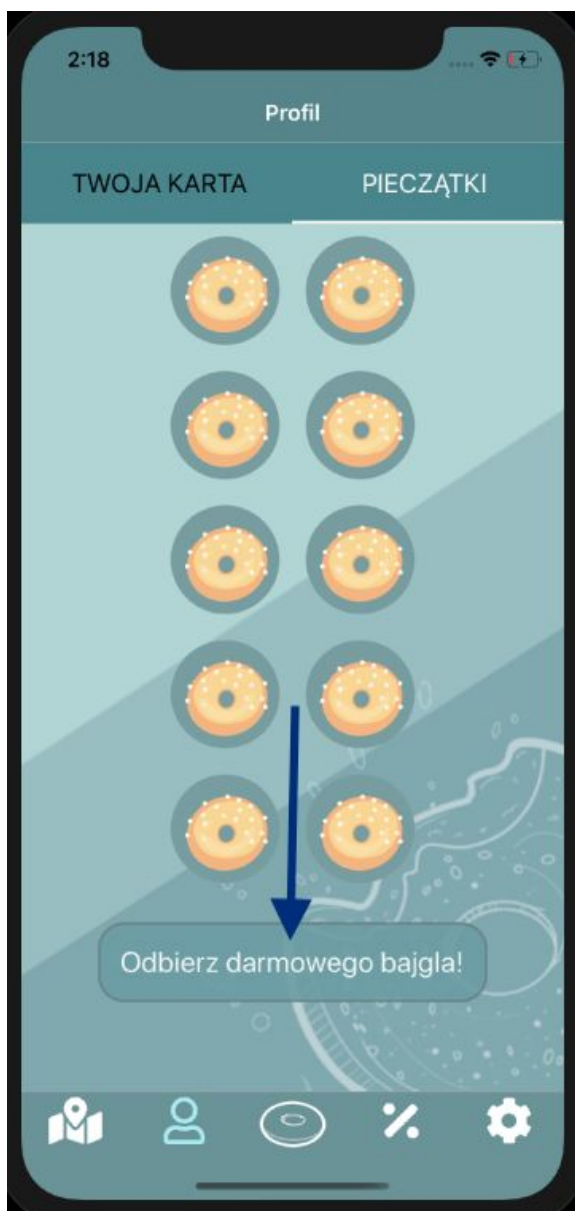
Profil - Twoja Karta

Kolejnym ekranem w aplikacji jest Profil. Znajdziesz go pod ikoną zaznaczoną kółkiem. Profil jest podzielony na **Twoją Kartę** oraz **Pieczałki**, do których nawigacja znajduje się u góry ekranu. **Twoja karta** to twój własny kod. Skanuj ją przy każdym zakupie a otrzymasz punkty oraz pieczątki.



Profil - Pieczątki

W zakładce [Pieczątki](#), znajdziesz ilość zebranych do tej pory pieczętek.



Profil - Pieczątki

Po zebraniu 10 pieczętek. Odblokuje się przycisk **Odbierz darmowego bajgla!** Nie musisz wykorzystać gratisu od razu, jednak dopóki nie odbierzesz nagrody, nie będą naliczane kolejne pieczątki



Profil - Pieczętki

Jeśli zdecydujesz się na odbiór nagrody. Udaj się do dowolnego punktu sprzedaży i pokaż sprzedawcy kod wygenerowany po kliknięciu **Odbierz darmowego bajgla!**



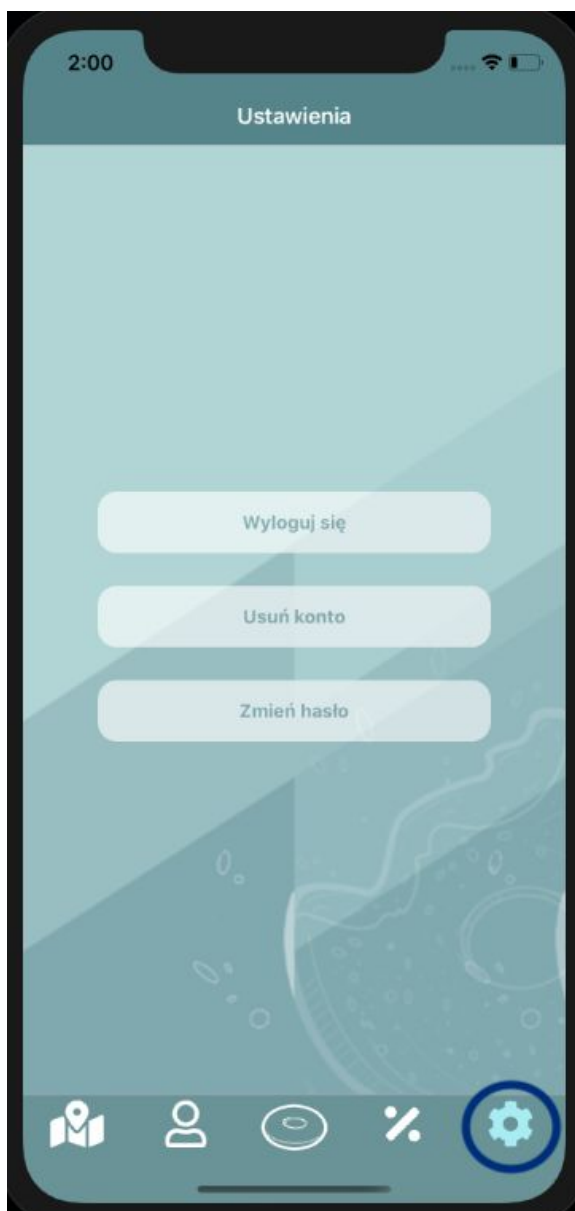
Promocje

Dostępne dla Ciebie promocje znajdziesz pod ikoną **Procent**. Są one generowane na podstawie zebranych przez Ciebie punktów.
Więcej punktów = więcej promocji



Promocje

Gdy zdecydujesz się, na którąś z dostępnych promocji kliknij [Pobierz QR](#) i pokaż wygenerowany kod sprzedawcy podczas zakupów. Po pobraniu nagrody zostaną odjęte punkty z twojego konta.



Ustawienia

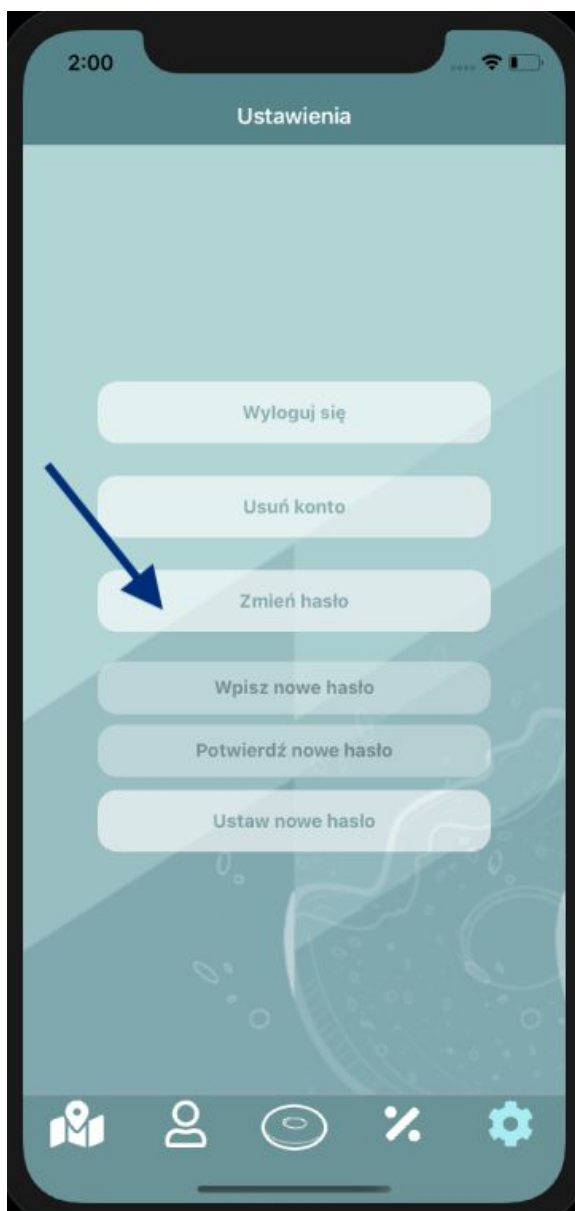
Ostatnim ekranem są Ustawienia.

Znajdują się tutaj 3 przyciski.

Wyloguj się - wylogowuje nas z aplikacji

Usuń konto - usuwa twoje konto, po usunięciu tracisz wszystkie zebrane nagrody, a aby ponownie korzystać z aplikacji musisz się zarejestrować

Zmień hasło



Ustawienia - Zmień Hasło

Po wybraniu **Zmień hasło** pojawią się 2 dodatkowe pola. Wpisz w nie **nowe hasło** i kliknij **Ustaw nowe hasło**, aby zapisać zmiany.

Aby mieć pewność, że nie pomyliłeś się wpisując hasło, zostaniesz poinformowany w przypadku różnic między **nowym hasłem**, a **potwierdzeniem**.

Lista wykorzystanych technologii

- React Native

React Native to cross-platformowy framework stworzony i rozwijany przez Facebooka, który umożliwia tworzenie aplikacji mobilnych z użyciem JavaScript na platformy iOS i Android. Pozwala współdzielić kod pomiędzy dwie platformy, dzięki zdolności do “tłumaczenia” JavaScript na natywne komponenty.

Na wybór tej technologii złożyły się głównie dwa czynniki: wcześniej wspomniana możliwość tworzenia aplikacji na platformy iOS i Android oraz szybszy proces developmentu niż w przypadku użycia natywnych języków. React Native pozwala na użycie funkcji “hot reloading”, która oszczędza czas poświęcany na kompilację kodu w przypadku użycia natywnych języków programowania. Dzięki temu zmiany wprowadzone w kodzie widoczne są znacznie szybciej (tj. po kilku sekundach).

- Spring

Spring jest platformą złożoną z wielu projektów, która dedykowana jest do tworzenia aplikacji w języku Java. Składa się z wielu osobnych modułów, z których w swoim projekcie możemy wybrać tylko te potrzebne.

Do wyboru tej technologii skłoniła nas możliwość wyboru modułów, które będą zawarte w naszej aplikacji. Dzięki temu może ona stać się dużo bardziej optymalna. Co więcej Spring jest frameworkiem, którego testowanie nie sprawia wielu problemów. Dostarcza wiele możliwości nieskomplikowanego wykonywania testów na wielu poziomach.

- MySQL

MySQL jest najbardziej popularnym open sourcowym systemem zarządzania relacyjną bazą danych.

Wybrałyśmy MySQL ponieważ jest bardzo wydajną bazą danych. Bez problemu poradzi sobie z wielkimi ilościami zapytań: może być w stanie obsługiwać nawet do miliona zapytań na sekundę. Co więcej wspiera format JSON.

Projekt testów

Testy oprogramowania to nieodłączny element podczas tworzenia aplikacji. Testowanie jest kluczowe, ponieważ pozwala sprawdzić czy kolejne zmiany w projekcie nie powodują błędów w starszych funkcjonalnościach, dlatego każda nowa funkcjonalność powinna być sprawdzona od razu po jej stworzeniu. Korzystamy zarówno z testów manualnych jak i automatycznych.

Testy manualne

Podejście manualne bazuje na zasobach ludzkich, pozwala dokładniej obserwować i ocenić, czy aplikacja jest łatwa. W testowaniu manualnym nie używa się frameworków. Wadą jest większe ryzyko wystąpienia błędu ludzkiego oraz czasochłonność.

Testy automatyczne

- **testy wydajnościowe:**
 - zbadanie czasu odpowiedzi serwera przy wzrastającej ilości jednoczesnych zapytań
- **Testy jednostkowe**
 - każdy element kodu sprawdzony w osobnym teście
- **Testy integracyjne**
 - testowanie poprawności komunikacji między bazą danych, a aplikacjami
 - testowanie poprawności komunikacji między aplikacją kliencką i aplikacją sprzedawcy
 - testowanie technicznych aspektów interfejsu (poprawna inicjalizacja, zapis/odbiór danych etc.)
- **Testy interfejsu użytkownika**
 - symulacja zachowania użytkownika
 - walidacja szczegółów (grafiki, kolory, fonty, poprawne wyświetlanie na różnych urządzeniach)

Wykorzystane narzędzia

- **junit** - narzędzie służące do tworzenia powtarzalnych testów jednostkowych oprogramowania napisanego w języku Java. Cechy JUnit: metoda najmniejszą jednostką testowania, przypadki testowe, oddzielenie testów od kodu, wiele mechanizmów uruchamiania, tworzenie raportów, integracja z różnymi środowiskami programistycznymi.
Użyty do testów backendu.
- **jest** - to framework do testowania JavaScript. Jest to platforma testowa służąca do rozwiązywania problemów w bazie kodu JavaScript. Jest w stanie niezawodnie uruchamiać testy równoległe, zapewniając jednocześnie jeden globalny stan testów.
Użyty do testów aplikacji w technologii React Native.
- **appium** - jest narzędziem open-source służącym do automatyzacji aplikacji natywnych, webowych oraz hybrydowych. Istotną cechą Appium jest wieloplatformowość.
Jest wykorzystane do testów UI.

Analiza ryzyka

Podczas szacowania ryzyka bierzemy pod uwagę :

- wpływ wystąpienia danego ryzyka
- prawdopodobieństwo wystąpienia danego zagrożenia

Lista ryzyk

np	Ryzyko	Prawdopodob.	Wpływ
	KORZYSTANIE Z OPROGRAMOWANIA		
1	Problemy ze sprzętem	Średnie	Duży
2	Błędy ludzkie	Średnie	Średni
3	Brak dostępu do internetu	Średnie	Duży
	PRACE PRZY PROJEKCIE		
4	Brak nadzoru z zewnątrz	Duże	Średni
5	Ograniczone zasoby czasowe	Duże	Duży
6	Zły harmonogram i preliminarz prac projektowo - wdrożeniowych	Małe	Średni
7	Nieumiejętność pracy zespołowej	Małe	Mały
8	Brak doświadczenia	Średnie	Średni
9	Realizacja prac niezgodna z ustaleniami	Małe	Średni
10	Niska jakość oprogramowania	Średnie	Duży
11	Niewłaściwy projekt GUI	Średnie	Średni
12	Źle zdefiniowane wymagania	Średnie	Duży
13	Zbyt optymistyczny harmonogram	Małe	Średni
14	Niewystarczające zasoby dostępne do wykonania pracy	Średnie	Duży
15	Czasochłonność wprowadzania i uzyskiwania danych	Średnie	Duży
16	Opóźnienia w przesyłaniu danych i ich przekłamania	Średnie	Duży

Macierz ryzyka

	Prawdopodobieństwo			
		Małe	Średnie	Duże
Wpływ	Mały	1	0	0
	Średni	3	3	1
	Duży	0	7	1

Sporo zagrożeń znalazło się w obszarze czerwonym (9 na 16), więc należy znaleźć rozwiązania, aby ryzyka te zlikwidować lub zmniejszyć (przesunąć na matrycy do obszaru zielonego / żółtego). Poniżej zostały przedstawione działania jakie podjął nasz zespół.

Sposoby reagowania na ryzyko

Ryzyko	Postępowanie
Brak nadzoru z zewnątrz	Konsultacje
Ograniczone zasoby czasowe	Planowanie swojego czasu.
Zły harmonogram i preliminarz prac projektowo - wdrożeniowych	Szczegółowe planowanie i nadzorowanie prac. Korzystanie z Trello.
Nieumiejętność pracy zespołowej	Ustalenie zasad i standardów współpracy .
Brak doświadczenia	Edukacja i szkolenie.
Realizacja prac niezgodna z ustaleniami	Konsultacje z prowadzącym. Wzajemne nadzorowanie swojej pracy.
Niska jakość oprogramowania	Zaplanowanie odpowiedniego czasu na testowanie i usuwanie błędów.
Niewłaściwy projekt GUI	Przeznaczenie wystarczająco dużo czasu na działania projektowe. Dokładne ustalenie wymagań przed rozpoczęciem projektowania.

Źle zdefiniowane wymagania	Opracowanie jasnych, kompletnych, szczegółowych, spójnych, osiągalnych i testowalnych wymagań, które są uzgodnione przez wszystkich członków grupy projektowej.
Zbyt optymistyczny harmonogram	Uwzględnienie odpowiedniego czasu na planowanie, projektowanie, testowanie, naprawianie błędów, ponowne testowanie, naprawianie błędów, ponowne testowanie, dokumentację, a także czasu wolnego od pracy (uwzględnić inne projekty).
Niewystarczające zasoby dostępne do wykonania pracy	Rozważenie alternatywnych metod, opcji.
Czasochłonność wprowadzania i uzyskiwania danych	Odpowiednie opracowanie architektury oprogramowania. Przemyślenie budowy bazy danych. Szukanie optymalnych rozwiązań.
Opóźnienia w przesyłaniu danych i ich przekłamania	Testowanie i poprawianie błędów.

Lista narzędzi użytych przy realizacji projektu

- IntelliJ - komercyjne zintegrowane środowisko programistyczne (IDE) dla Javy firmy JetBrains.
- Visual Studio Code - edytor programistycznych kodów źródłowych, stworzony przez Microsoft. W naszym przypadku użyty do tworzenia aplikacji mobilnych.
- Trello - aplikacja webowa firmy Atlassian umożliwiająca tworzenie tablic kanbanowych.
- Adobe Illustrator - rozbudowany program graficzny przeznaczony do tworzenia i edycji grafiki wektorowej, będący jednym z flagowych produktów firmy Adobe Inc.
- GitHub - hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git, produkt firmy Microsoft.