Dokumentacja

Tytuł: “Aplikacja mobilna do wyszukiwania ofert lodziarni rzemieślniczych”
Autor: Tomasz Boczarski
Przedmiot: Programowanie aplikacji mobilnych

Cel: Implementacja aplikacji mobilnej służącej do wyszukiwania ofert lodziarni rzemieślniczych.

Wymagania funkcjonalne

- Wyszukiwanie lodziarni w określonej odległości od aktualnej lokalizacji użytkownika.
- Wyszukiwanie lodziarni w podanym przez użytkownika mieście.
- Dodawanie wyszukanych lodziarni do ulubionych.
- Wyświetlanie lodziarni dodanych do ulubionych.
- Ustalanie promienia w kilometrach w jakim ma wyszukiwać lodziarnie.
- Wyświetlanie informacji o lodziarni - nazwa, logo, adres, dostępne smaki, dodatkowe informacje.
- Pokazanie lokalizacji lodziarni w Google Maps

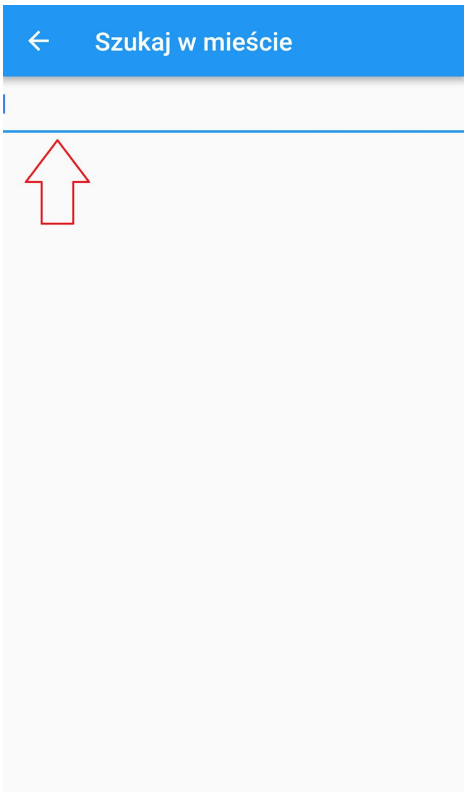
Wymagania niefunkcjonalne

- Implementacja w języku programowania *dart* przy użyciu szkieletu aplikacji *flutter*
- Dane lodziarni pobierane z serwera

Specyfikacja zewnętrzna

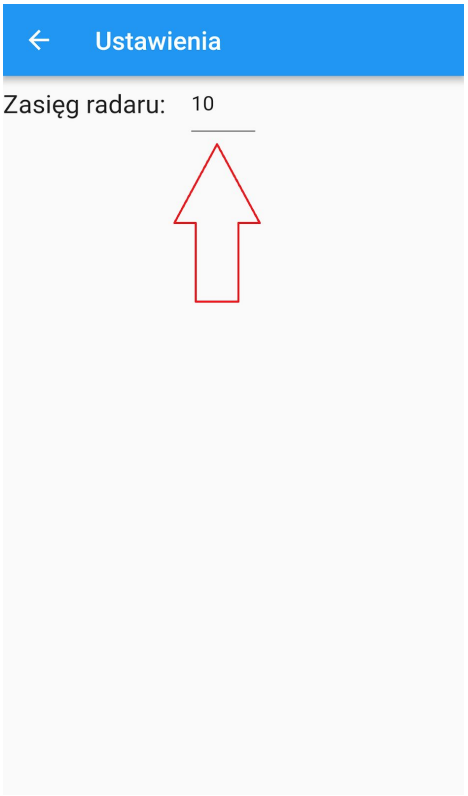
Menu główne

1. Przejście do widoku wyszukiwania w okolicy.
2. Przejście do widoku wyszukiwania w mieście.
3. Przejście do widoku wyświetlania listy lodziarni dodanych do ulubionych.
4. Przejście do widoku ustawień.



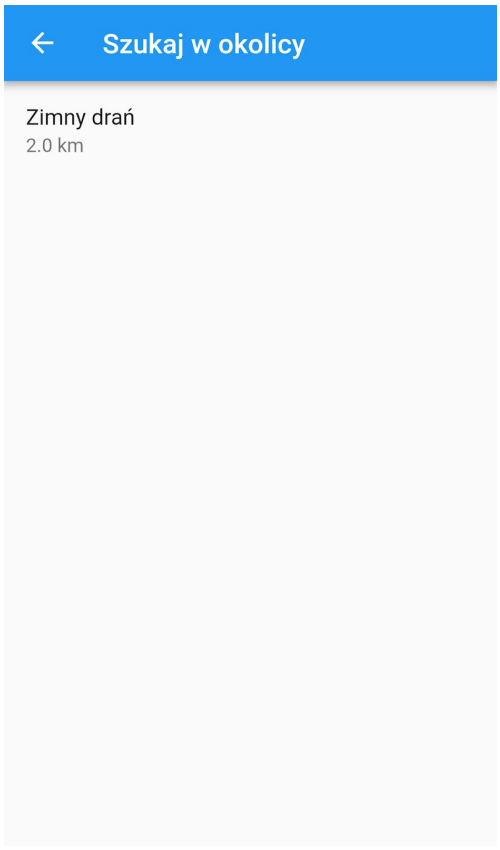
Widok wyszukiwania w mieście

Aby wyszukiwać lodziarnie w mieście należy w zaznaczone miejsce wprowadzić nazwę miasta i zatwierdzić.



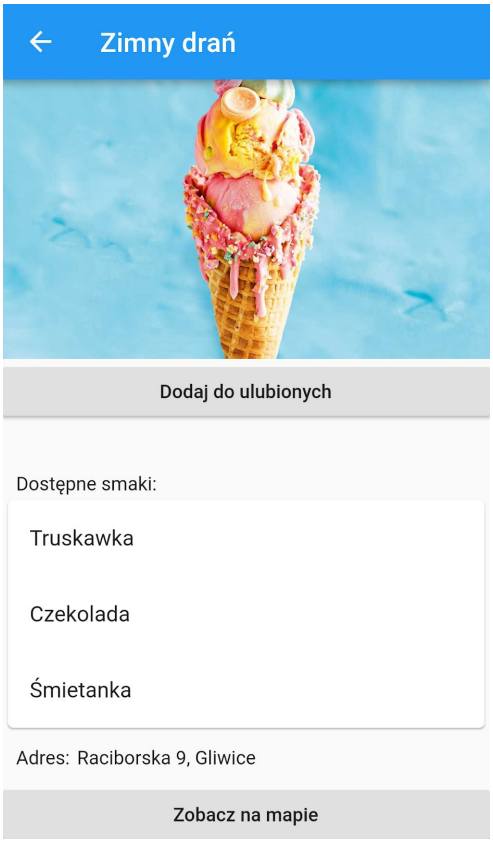
Widok ustawień

Aby zmienić zasięg wyszukiwania w okolicy należy w ustawieniach wprowadzić w zaznaczonym miejscu liczbę kilometrów w promieniu których mają być wyszukiwane lodziarnie



Widok wyszukiwania w okolicy/listy dodanych do ulubionych

Aby odświeżyć listę należy przeciągnąć listę w dół.
Aby przejść do szczegółów lodziarni należy wybrać element z listy.



Widok szczegółów lodziarni

Aby dodać lodziarnie do ulubionych należy nacisnąć “Dodaj do ulubionych”
Przycisk “Zobacz na mapie” przekieruje nas do map google z zaznaczoną lokalizacją lodziarni

Specyfikacja wewnętrzna

- **Struktura aplikacji**
  Oparta o widżety. Wyróżniamy widżety posiadające stan oraz nie posiadające stanu.

- **Użyte biblioteki zewnętrzne**
  - *location* - do pozyskania lokalizacji z modułu gps urządzenia
  - *url_launcher* - do przekierowania do google maps
  - *http* - do komunikacji z serwerem
  - *latlong* - do obliczania odległości
  - *shared_preferences* - do zapisu ustawień i ulubionych lodziarni

- **Klasy**
  - **MyApp**
    - Klasa opisująca główny widżet w którym znajduje się aplikacja
  - **ICSMenu**
    - Klasa opisująca widżet widoku menu
  - **ICSScanArea**
    - Klasa opisująca widżet widoku szukania w okolicy
    - **Funkcje**
      - **initState()** - funkcja ustawiająca stan początkowy
  - **ICSCitySearch**
    - Klasa opisująca widżet widoku szukania w mieście
    - **Funkcje**
      - **initState()** - funkcja ustawiająca stan początkowy
  - **ICSFavoritesList**
    - Klasa opisująca widżet widoku wyświetlania listy ulubionych lodziarni
    - **Funkcje**
      - **initState()** - funkcja ustawiająca stan początkowy
  - **ICSSettings**
    - Klasa opisująca widżet widoku ustawień
    - **Funkcje**
      - **initState()** - funkcja ustawiająca stan początkowy
  - **ICSDetails**
    - Klasa opisująca widżet widoku szczegółów lodziarni
    - **Funkcje**
      - **initState()** - funkcja ustawiająca stan początkowy
      - **showOnMap()** - funkcja powodująca przekierowanie do map google
      - **toggleFavorites()** - funkcja zapisująca/usuwająca do/z ulubionych
      - **getFavoriteStatus()** - funkcja zwracająca odpowiedni tekst w zależności od tego czy lodziarnia została dodana do ulubionych bądź nie

                - **isFavorite()** - funkcja zwracająca wartość *true* jeśli lodziarnia jest dodana do ulubionych i *false* w przeciwnym wypadku
        - **IcecreamShopsService**
            - Klasa implementująca wzorzec singleton, służąca do pobierania danych lodziarni.
            - **Funkcje**
                - **scanArea()**
                    - funkcja zwracająca listę lodziarni znajdujących się w zasięgu
                - **citySearch(String city)**
                    - funkcja zwracająca listę lodziarni znajdujących się w mieście podanym w parametrze *city*
                - **getFavorites()**
                    - funkcja zwracająca listę lodziarni dodanych do ulubionych
        - **IcecreamShop**
            - Klasa opisująca obiekt lodziarni
            - **Pola**
                - int id;
                - String name;
                - Address address;
                - String logoUrl;
                - String additionalInfo;
                - List<dynamic> flavours;
                - double distance;
            - **Funkcje**
                - IcecreamShop.fromJSON(parsedJSON) - funkcja zwracająca obiekt z przeanalizowanego obiektu JSON
        - **Address**
            - klasa opisująca obiekt adresu lodziarni
            - **Pola**
                - String street;
                - String city;
                - double latitude;
                - double longitude;
            - **Funkcje**
                - Address.fromJSON(parsedJSON) - funkcja zwracająca obiekt z przeanalizowanego obiektu JSON
                - toString() - funkcja zwracająca adres w postaci obiektu typu String
- **Klasy z pakietu flutter**
    - **MaterialApp**
        - Klasa podstawowa każdej aplikacji
    - **Scaffold**
        - Klasa stanowiąca podstawowy szkielet widoku w aplikacji

    - **Center**
        - Klasa opakowująca widżet w celu wyśrodkowania go
    - **Column**
        - Klasa opakowująca listę widżetów ustawiająca zawartość w kolumnie
    - **Row**
        - Klasa opakowująca listę widżetów ustawiająca zawartość w wiersz
    - **Container**
        - Klasa opakowująca widżet pozwalająca na dodanie dekoratorów np cieni
    - **Image**
        - Klasa służąca do wstawiania obrazów
    - **RawMaterialButton**
        - Klasa opisująca widżet guzika
    - **RaisedButton**
        - Klasa opisująca widżet guzika typu *raised*
    - **MaterialPageRoute**
        - Klasa pozwalająca na przełączanie widoków
    - **AppBar**
        - Klasa opisująca pasek na górze aplikacji
    - **RefreshIndicator**
        - Klasa umożliwiajaca wywołanie funkcji *onRefresh* która wywołuje się przy przeciagnięciu listy w dół
    - **Text**
        - Klasa opakowująca wartość typu String jako widżet
    - **Card**
        - Klasa opakowująca widżet wyświetlając jego zawartość we wnętrzu karty
    - **ListView**
        - Klasa opakowująca listę widżetów które mają wyświetlać się w postaci listy
    - **ListTile**
        - Klasa opakowująca zawartość pojedynczego elementu listy w ListView
- **Serwer**
    - **Punkty końcowe**
        - **/icecreamshops GET**
            - Zwraca listę obiektów spełniających warunki parametrów
            - **Dozwolone parametry**
                - **lat** - szerokość geograficzna punktu od którego ma przeszukiwać okolicę
                - **long** - długość geograficzna punktu od którego ma przeszukiwać okolicę
                - **rad** - promień przeszukiwania
                - **city** - nazwa miasta

- **Odpowiedź**

```
[
  {
    "id": Integer,
    "name": String,
    "logoUrl": String,
    "additionalInfo": String,
    "address": {
      "city": String,
      "street": String,
      "latitude": double,
      "longitude": double
    },
    "flavours": String[]
  }
]
```

- **/favorites POST**
  - Zwraca listę obiektów których id są podane w ciele zapytania
  - **Odpowiedź**

```
[
  {
    "id": Integer,
    "name": String,
    "logoUrl": String,
    "additionalInfo": String,
    "address": {
      "city": String,
      "street": String,
      "latitude": double,
      "longitude": double
    },
    "flavours": String[]
  }
]
```

- **Wnioski**
  - Zagnieżdżenie obiektów w notacji JSON było problematyczne. Uproszczenie poprzez usunięcie zbędnych zagnieżdżeń i implementacja fabryk *fromJSON* w obiektach rozwiązało problem.