

# Lista witamin wraz z wyszukiwarką objawów

## 1. Cel projektu

Celem projektu było stworzenie aplikacji zawierającej listę witamin wraz z objawami ich nadmiaru, niedoboru oraz występowaniem w pożywieniu, a także umożliwienie użytkownikowi wpisania w wyszukiwarkę symptomu i na tej podstawie wyświetlenie spisu witamin, które zawierają podany objaw.

## 2. Funkcjonowanie aplikacji

Aplikacja po uruchomieniu daje użytkownikowi wgląd w listę dostępnych witamin oraz dostęp do wyszukiwarki objawów. Wpisanie interesującego nas hasła filtruje listę ograniczając ją do spisu witamin, w której dany objaw nadmiaru lub niedoboru się znajduje. Po takim wyszukiwaniu aplikacja dalej wyświetla listę, do której elementów mamy dostęp. Wejście w którąś z witamin skutkuje wyświetleniem listy objawów oraz artykułów spożywczych bogatych w konkretną substancję. Powyższa procedura może być wielokrotnie powtarzana bez konieczności restartu całej aplikacji czy bez żadnych dodatkowych kroków z wyjątkiem użycia ikony powrotu.

## 3. Wykorzystana architektura

- Plik w formacie JSON dołączony do Android Studio w katalogu *assets*
- Parsowanie JSON (funkcja *Utils.kt*, umożliwiającą dostęp do danych zawartych w pliku znajdującym się w *assets*; biblioteka *Gson*, służąca do przekonwertowania danych z JSON na obiekty klasy)
- Klasa typu *Parcelable*: *Vitamins.kt* (zawierająca cztery atrybuty: *id* (String), *objawy\_nadmiaru* (ArrayList<String>), *objawy\_niedoboru* (ArrayList<String>), *wystepowanie* (ArrayList<String>); trzy metody: *writeToParcel*, *describeContents*, *newArray*; *companionobject:CREATOR*)
- RecyclerView do wyświetlenia listy witamin
- Klasy do obsługi RecyclerView: *VitaminsSelectionViewHolder.kt* (zawierająca jeden atrybut: *vitaminName*) oraz *VitaminsSelectionRecyclerViewAdapter.kt* (zawierająca jeden atrybut: *vitaminsNames*; trzy metody: *getItemCount*, *onBindViewHolder*, *onCreateViewHolder*; interfejs: *VitaminSelectionRecyclerViewClickListener*)
- Aktywność umożliwiającą wyświetlenie szczegółów witaminy: *activity\_vitamins\_details.xml*
- Cztery TextView (pole layout\_constraintTop\_toBottomOf zawiera id poprzedzającego TextView) do wyświetlania nazwy witaminy, objawów jej nadmiaru i niedoboru oraz występowania w pożywieniu
- Klasa *Intent* do wyświetlenia detali witaminy
- *companionobject:INTENT\_VITAMIN\_KEY*
- Klasa do wyświetlania szczegółów witaminy: *VitaminsDetails.kt* (zawierająca pięć atrybutów: *vitamin*, *idTextView*, *nadmiarTextView*, *niedoborTextView*, *wystepowanieTextView*; jedną metodę: *onCreate*)
- Element menu odpowiedzialny za wyszukiwanie: *menu\_search*
- Klasa *MainActivity.kt* zawierająca metody: *OnCreate*, *onOptionsItemSelected*, *showVitaminDetail*, *vitaminItemClicked* oraz *onCreateOptionsMenu*, w której w przypadku wyszukiwania funkcja *onQueryTextChange* tworzy listę witamin zawierających wyszukiwane słowo w objawach i przekazuje ją do adaptera RecyclerView

## 4. Problemy

Pierwszy problem stanowiło to, w jaki inny sposób niż ręcznie dodać do aplikacji listę witamin. Rozwiązano to poprzez zapisanie parametrów witamin w formacie JSON i wykorzystanie parsowania JSON do obiektów klasy. Przy wykorzystaniu funkcji *toLowerCase()* udało się wyeliminować kolejny problem - znaczenie wielkości wpisywanych liter.