

Dokumentacja projektu

„Shopping List”

Przedmiot: Programowanie urządzeń mobilnych

Autor: Dariusz Kowal

O projekcie

Tematem projektu jest mobilna lista zakupów czyli „Shopping List”, która ułatwia robienie zakupów i odciąża naszą pamięć poprzez organizacje tej listy.

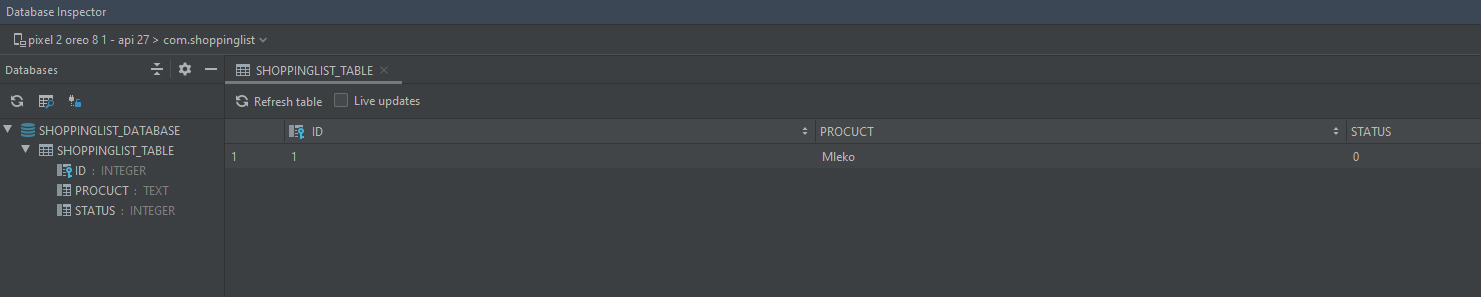
Głównym zadaniem było stworzenie aplikacji mobilnej cechującej się podstawowymi wersji CRUD. Projekt ten został stworzony jako praca zaliczeniowa na przedmiot „Programowanie urządzeń mobilnych”

Technologie

Jako bazę danych wykorzystałem SQLite – jest to bezserwerowa, relacyjna, lekka baza danych, którą bardzo często wykorzystuje się przy tworzeniu aplikacji na iOS’a i Android’a. Interfejs graficzny w formie aplikacji mobilnej powstał przy użyciu programu Android Studio.

Baza danych

SQLite jest idealna dla relacyjnej bazy danych znajdującej się przy aplikacji. Dane przechowywane są lokalnie, czyli nie potrzeba mieć dostępu do internetu aby sprawdzić naszą listę zakupów. Zawartość bazy danych przetrzymywana jest w jednym pliku.



Rysunek 1. Struktura bazy danych

Baza danych składa się z kolumn i wierszy.

-ID – ID produktu

-procuct – produkt (przy tworzeniu powstała literówka)

-status – status, który odpowiada czy produkt został kupiony czy też nie

Struktura aplikacji

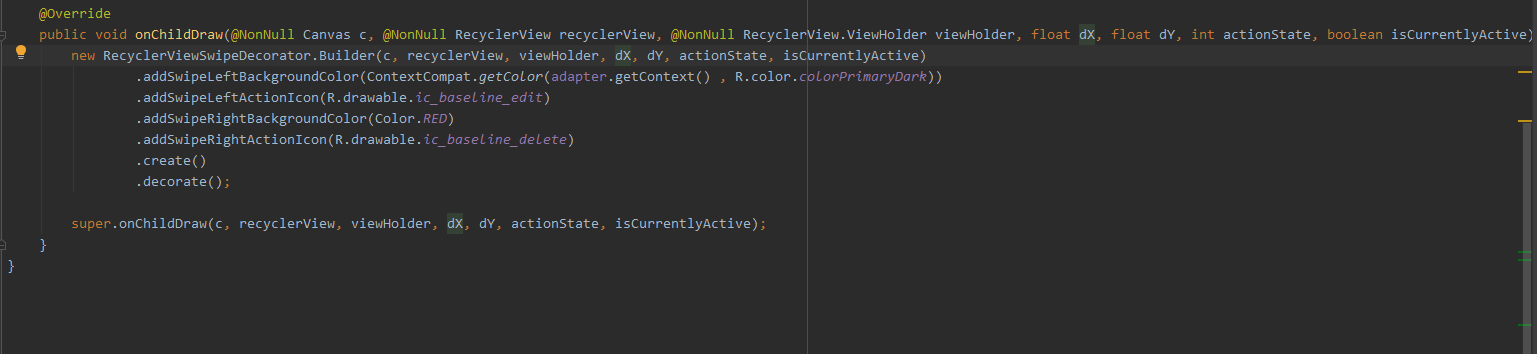
|  |  |
| --- | --- |
| Struktura projektu jest typowa dla aplikacji napisanych w Javie w Android Studio.  W package’u Adapter klasa ShoppingListAdapter odpowiada za pobieranie danych z zestawu i generowanie obiektów View opartych o te dane.  W package’u Model klasa ShoppingListModel odpowiada za kontrolowanie jak dane są tworzone, składowane i modyfikowane.  W package’u Utils klasa DataBaseHelper odpowiada za tworzenie/aktualizowanie struktury bazy danych.  Klasa AddNewProduct zawiera metody onCreateView, onViewCreated  Klasa MainActivity zawiera metody onCreate oraz onDialogClose.  Interfejs OnDialogCloseListener odpowiada za okna dialogowe.  Klasa RecyclerViewTouchHelper odpowiada za edycję oraz usuwanie produktów.  Klasa SplashActivity odpowiada za ekran ładowania.  W folderze drawable są vector assety. Są to ikony usuwania/edytowania itp.  W folderze layout jest nasz cały układ aplikacji zapisany z rozszerzeniem .xml  W folderze values :  - plik colors.xml zawiera zdefiniowane kolory, które są używane do stylowania aplikacji.  - plik string.xml zawiera zdania, które są używane w aplikacji  - plik themes.xml odpowiada za motyw aplikacji | Rysunek 2. Struktura projektu |

Przykład kodu



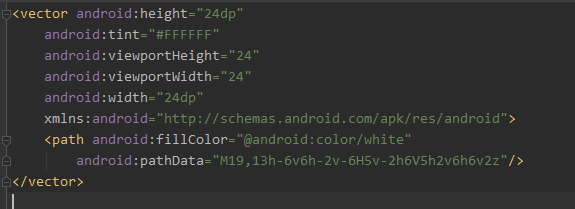
Rysunek 3. Tworzenie/edycja/usuwanie produktu oraz aktualizacja statusu produktu

Powyższy fragment kodu przedstawia klasę „DataBaseHelper” która odpowiedzialna jest za łączenie z bazą danych/edycją/dodawaniem oraz usuwaniem danych. Na samej górze pliku mamy zdefiniowaną metodę „insertProduct”. Polega ona na stworzeniu produktu o nazwie, którą podamy z klawiatury i dodanie jej do bazy danych.

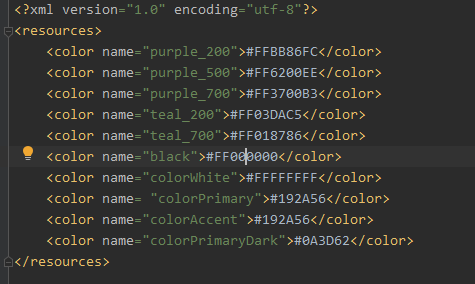


Rysunek 4. Metoda klasy RecyclerViewTouchHelper

Jest to prosta klasa narzędziowa do dodawania tła, ikony i etykiety do elementu RecyclerView podczas przesuwania w prawo lub w lewo.

Rysunek 5. Plik ic\_baseline\_add\_24.xml Rysunek 6. Design pliku ic\_baseline\_add\_24.xml



Rysunek 7. Plik colors.xml

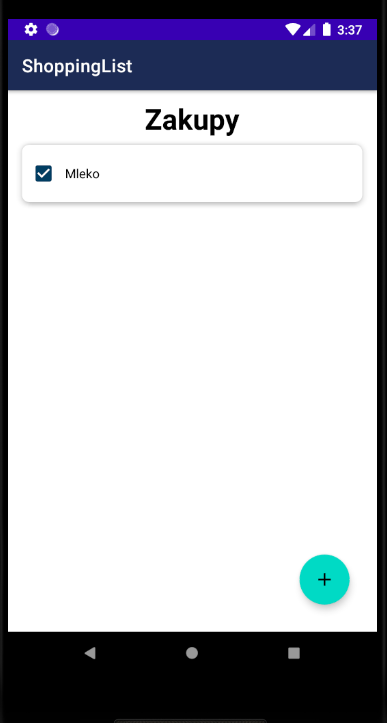
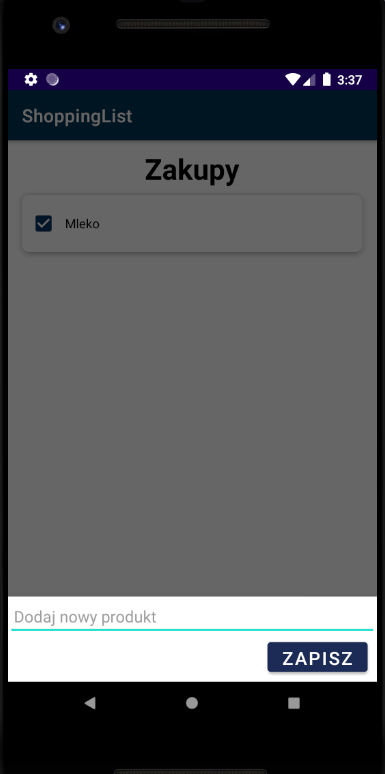
W pliku „Colors” zdefiniowane są wszystkie kolory użyte w aplikacji. Dzięki temu, możemy w jednym miejscu zmienić dany kolor, aby jego zastosowanie było widoczne w całej aplikacji. Pozwala nam to na zaoszczędzenie czasu oraz skrócenie kodu.

Prezentacja aplikacji

|  |  |
| --- | --- |
| Rysunek 8.Widok startowy | Rysunek 9. Menu główne do tworzenia zakupów |

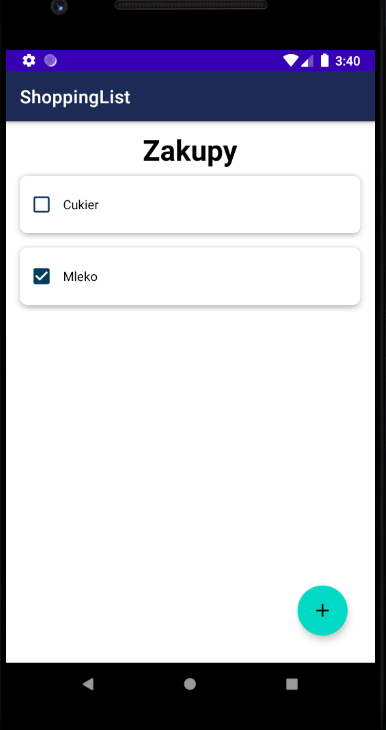
Po uruchomieniu aplikacji naszym oczom ukaże się widok startowy aplikacji (rys. 8)

W prawym dolnym rogu znajduje się przycisk „z plusem” aby dodać nowy produkt. Możemy także kliknąć na „Mleko” aby zaznaczyć ten produkt.

|  |  |
| --- | --- |
| Rysunek 10. Zaznaczanie kupionego produktu | Rysunek 11. Tworzenie nowego produktu |

Po kliknięciu na przycisk dodawania Ukazuje nam się pole tekstowe do pisania oraz przycisk zapisz (rys. 11).

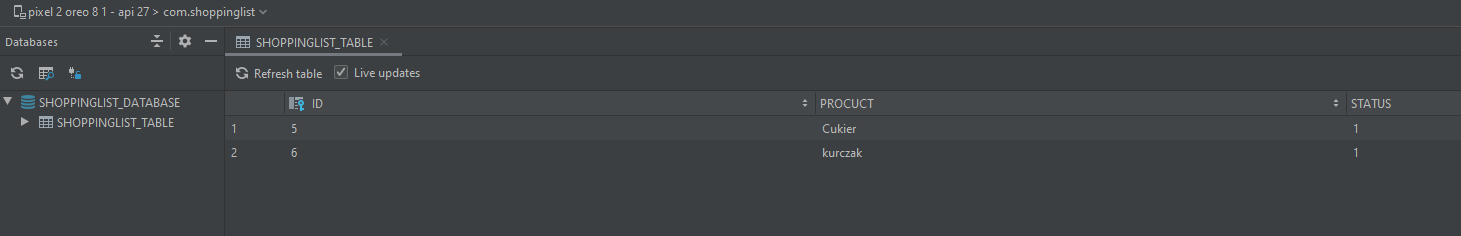


Rysunek 11. Utworzony produkt wyświetla się jako pierwszy na liście

Po zapisaniu produktu pojawia się nam na górze listy, aby łatwiej było znaleźć produkty zapisane w ostatniej chwili przed zakupami.

|  |  |
| --- | --- |
| Rysunek 12.Edycja dodanych produktów | Rysunek 13. Usuwanie produktów z listy |

Po przesunięciu w lewo produktu, możemy go edytować (pojawi nam się odpowiednia ikona oraz kolor tak jak na rys.12). Gdy chcemy usunąć produkt należy przesunąć produkt w prawo. Dostaniemy komunikat czy na pewno chcemy usunąć dany produkt (pojawi nam się odpowiednia ikona oraz kolor tak jak na rys.13).



Rysunek 14. Baza danych po usunięciu mleka i dodaniu kurczaka oraz zaznaczeniu ich jako kupione.

Rysunek 14 pokazuje nam w czasie rzeczywistym zmiany w aplikacji.

Uruchamianie projektu

Projekt można pobrać tutaj: https://github.com/KowDar/PUM\_SHOPPINGLIST

Jeżeli chcemy uruchomić aplikację musimy mieć zainstalowane środowisko Android Studio. W Android Studio powinniśmy mieć zainstalowane urządzenie mobilne lub podpięty własny telefon przez kabel USB. W tym przypadku z co najmniej API 27 tj. wersja Androida Pie 8.1. Po zaimportowaniu projektu automatycznie powinien zainstalować się gradle.