Zespół Szkół NR.10

Im. Stanisława Staszica

Temat: Sprawozdanie z aplikacji ***Vinted***

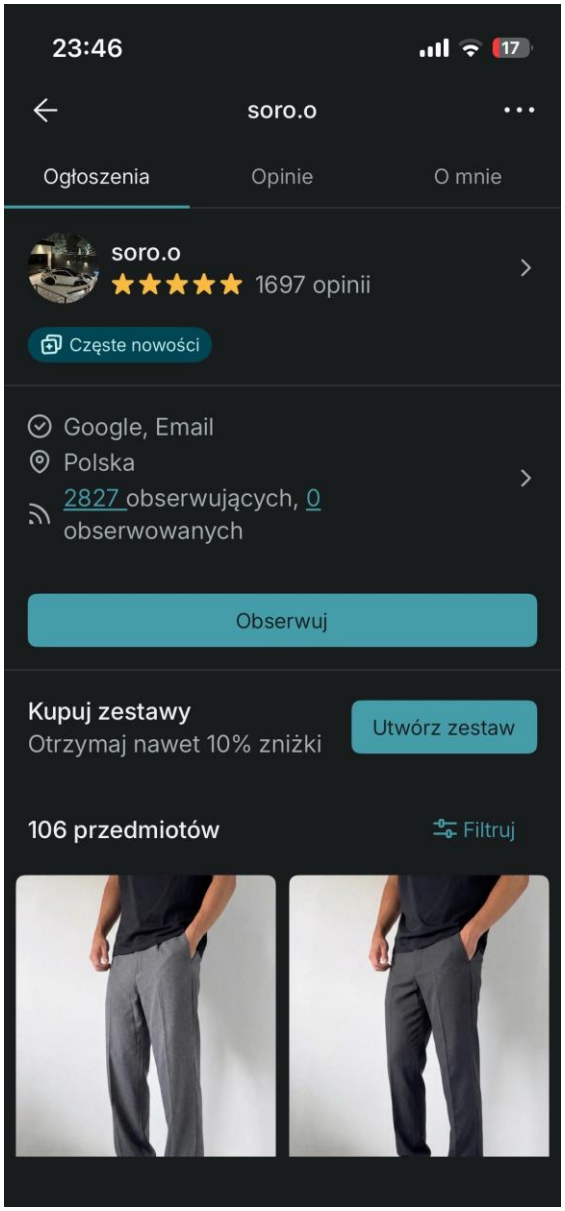
Autor: Antoni Paczkowski
Warszawa, 13.01.2026 r.

W ramach tego sprawozdania wybrałem aplikację Vinted, która jest popularną platformą do kupna i sprzedaży używanych ubrań.

1. Widok profilu sprzedawcy (np. profil użytkownika "soro.o" z opiniami, obserwującymi i listą przedmiotów).
2. Widok profilu użytkownika (np. własny profil z odznakami, portfelem i opcjami nawigacji).
3. Widok dodawania przedmiotu (formularz do tworzenia nowej oferty sprzedaży).
4. Widok wyszukiwania (ekran wyszukiwania przedmiotów lub z historią zapytań).

1. Widok profilu sprzedawcy

Ten widok wyświetla informacje o sprzedawcy, takie jak nazwa użytkownika, ocena, liczba opinii, lokalizacja, liczba obserwujących i obserwowanych, przycisk do obserwowania oraz listę dostępnych przedmiotów (w formie miniaturek zdjęć). Struktura jest pionowa, z nagłówkiem profilu na górze i scrollowalną listą przedmiotów poniżej. Użytkownik może przeglądać oferty, kliknąć w przedmiot, aby zobaczyć szczegóły, lub obserwować sprzedawcę. Każdy przedmiot w liście działa jak osobny przycisk, bo po kliknięciu pokazuje szczegóły oferty, co czyni całą listę pełną interaktywnych elementów nawigacyjnych.



Lista kontrolek:

- ScrollView: Jest to kontener przewijalny, obejmujący cały widok, aby umożliwić scrollowanie długiej treści, takiej jak lista przedmiotów; znajduje się na najwyższym poziomie hierarchii.
- LinearLayout (pionowy): Służy jako główny kontener wewnątrz ScrollView, układający elementy pionowo od góry do dołu; obejmuje nagłówek i sekcje poniżej.
- RelativeLayout: Używany w nagłówku profilu do pozycjonowania elementów względem siebie, np. obrazu obok tekstów; znajduje się na górze LinearLayout.
- ImageView: Wyświetla zdjęcie profilowe sprzedawcy; umieszczone po lewej stronie w RelativeLayout nagłówka.

- TextView (kilka razy użyte): Jedna dla nazwy użytkownika (duży, pogrubiony tekst), inna dla oceny z gwiazdkami i liczbą opinii, oraz jedna dla liczby obserwujących/obserwowanych; wszystkie umieszczone obok ImageView w RelativeLayout, ułożone pionowo pod sobą.
- Button: Przycisk do obserwowania sprzedawcy; znajduje się po prawej stronie nagłówka w RelativeLayout.
- TextView: Dodatkowy tekst informujący o zniżkach na zestawy; umieszczony poniżej nagłówka w LinearLayout.
- ListView: Lista przedmiotów na sprzedaż, wyświetlająca miniaturki i opisy, gdzie każdy element listy działa jak przycisk do szczegółów; znajduje się na dole LinearLayout, zajmując resztę przestrzeni.
- Button (kilka razy użyte, w elementach ListView): Każdy przedmiot w liście to w zasadzie przycisk z obrazem i tekstem, ukryty w strukturze ListView, ale klikalny; rozmieszczone w każdym wierszu listy.

Opis funkcjonalności w Kotlinie:

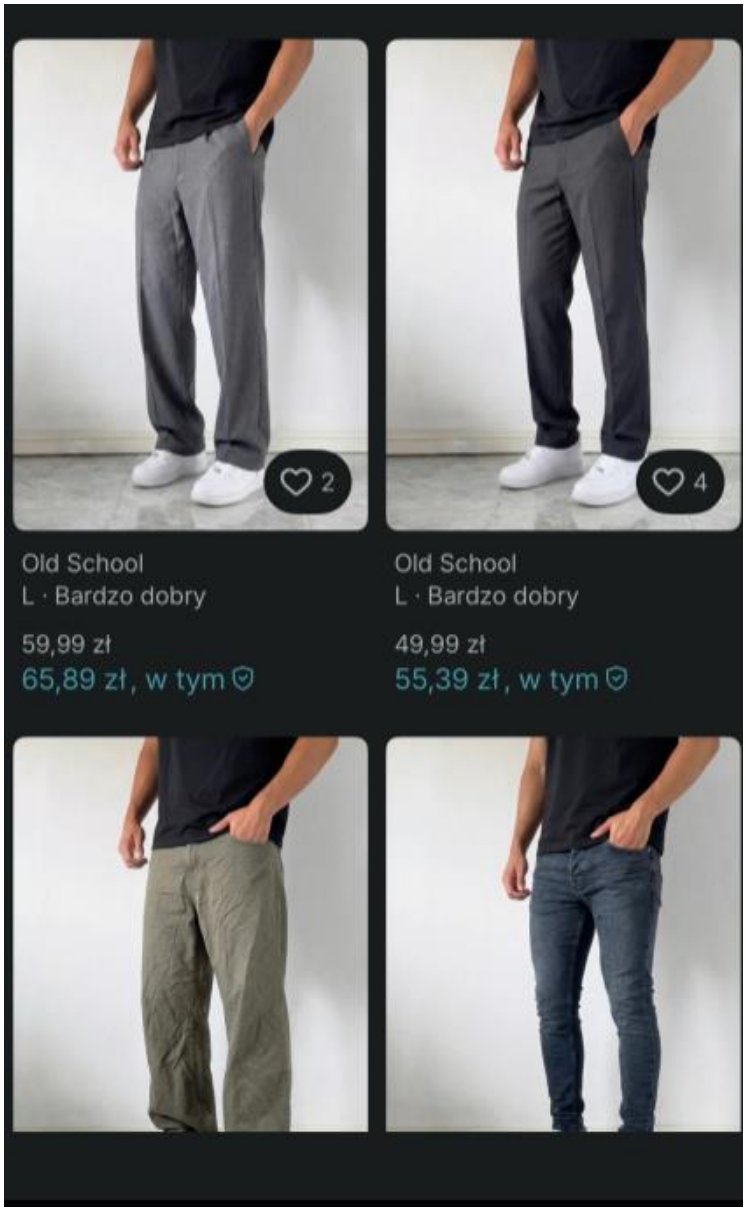
Dla ImageView: imageView.setImageURI(). Komponent mógłby pokazywać proste zdjęcie z internetu, ustawione przez użytkownika

Dla TextView username_text: textView.text = "nazwa". Te elementy mogłyby wyświetlać proste teksty o sprzedawcy, które zamieścił o sobie

Dla Button (follow_button): button.setOnClickListener { }. Przycisk mógłby po dotknięciu zmienić swój tekst, na przykład na "obserwujesz", i powiedzieć aplikacji, żeby zapisała tę zmianę, aby użytkownik dostawał powiadomienia od sprzedawcy.

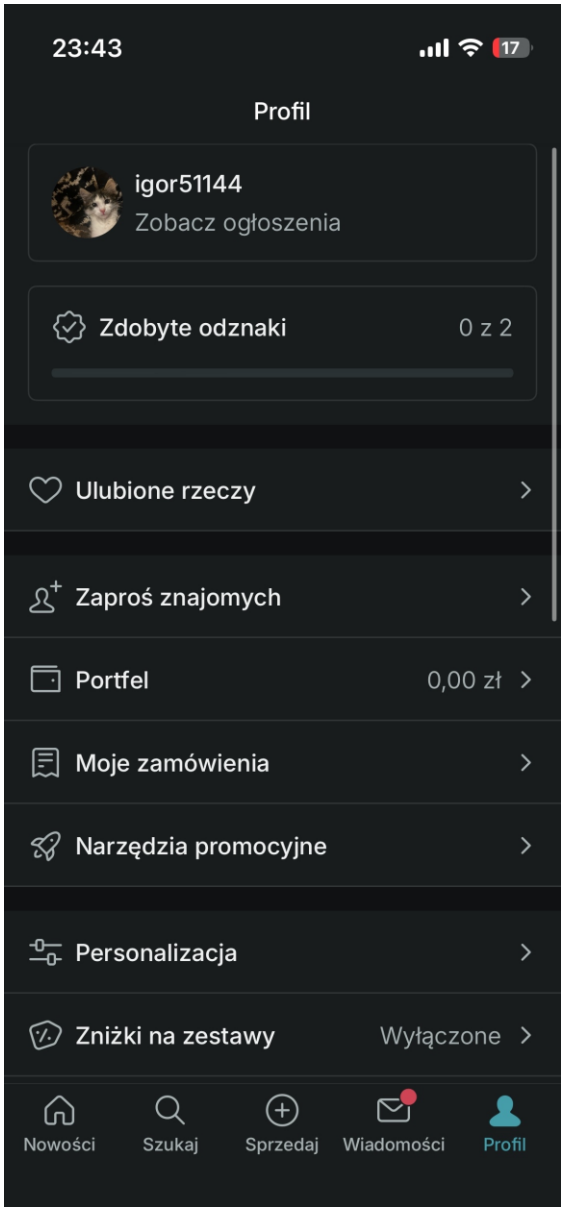
Dla ListView Lista mogłaby pokazywać proste elementy z przedmiotami, a po dotknięciu jednego przenosić do innej strony z więcej szczegółami, i ładować więcej rzeczy, gdy użytkownik scrolluje w dół.

Dla Button (w elementach ListView): button.setOnClickListener { }. Każdy taki przycisk w liście mógłby po dotknięciu otwierać stronę ze szczegółami przedmiotu, pokazując więcej zdjęć czy opisów, i pozwalać na kupno lub dodanie do ulubionych.



2. Widok profilu użytkownika

Ten widok pokazuje własny profil użytkownika, z elementami takimi jak nazwa, awatar, zdobyte odznaki, ulubione rzeczy, portfel, zamówienia, narzędzia promocyjne i personalizacja. Jest to lista opcji nawigacyjnych, z wartościami numerycznymi (np. saldo portfela 0.00 zł). Użytkownik może klikać w sekcje, aby przejść dalej. Każda opcja typu "Portfel" czy "Zamówienia" działa jak przycisk, przenoszący do innego widoku, co czyni cały profil pełen interaktywnych elementów do nawigacji.



Lista kontrolek:

- ScrollView: Kontener przewijalny dla całej listy opcji, aby umożliwić scrollowanie jeśli sekcji jest dużo; na najwyższym poziomie.
- LinearLayout (pionowy): Główny układ pionowy wewnątrz ScrollView, grupujący nagłówek i sekcje opcji.
- ImageView: Wyświetla awatar użytkownika
- TextView: Pokazuje nazwę użytkownika; obok ImageView
- LinearLayout (poziomy): cały klikalny jak przycisk; ułożone pionowo w głównym LinearLayout.
- ImageView (kilka instancji): Ikony dla każdej sekcji, np. odznaka czy serce; po lewej w każdym sekcyjnym LinearLayout.
- TextView (kilka instancji): Etykiety sekcji (np. "Zdobyte odznaki") i wartości (np. "0 z 2"); w środku i po prawej w sekcyjnych LinearLayout.

- Button (kilka razy, w sekcjach): Każda opcja jak "Portfel" to w zasadzie przycisk wewnątrz LinearLayout, umożliwiający przejście do innych widoków, rozmieszczone w każdej sekcji listy opcji.

Opis funkcjonalności w Kotlinie:

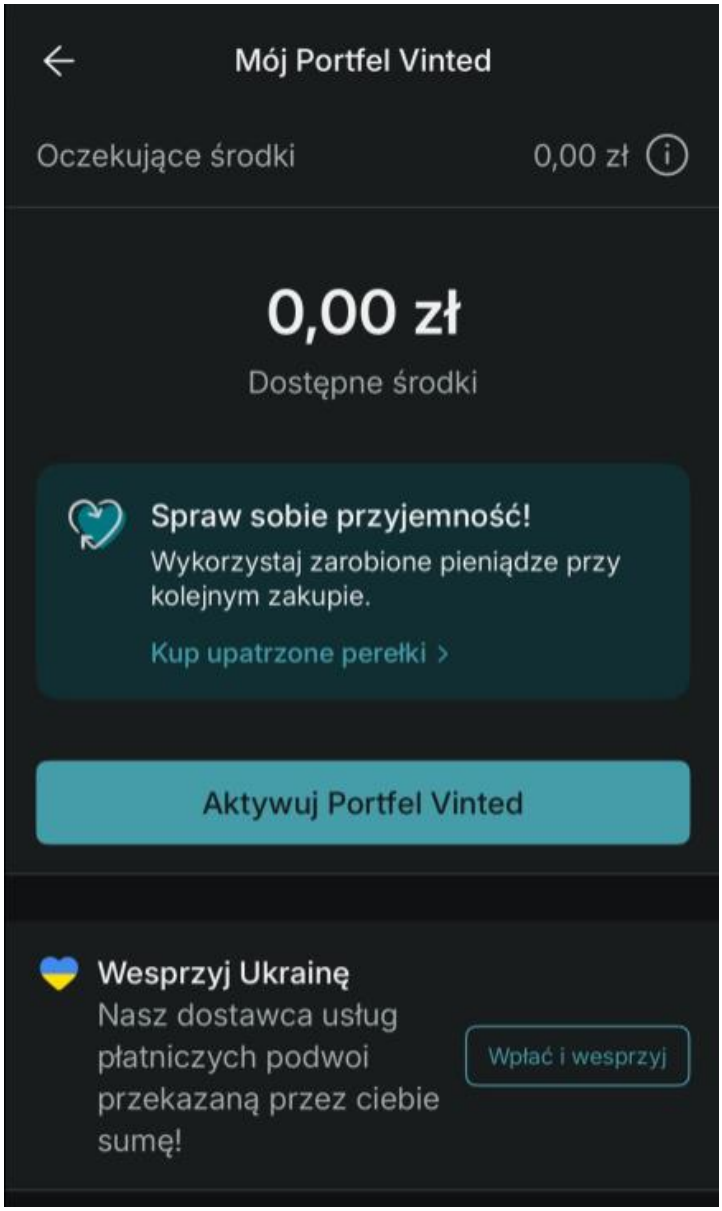
Dla ImageView (user_avatar): imageView.setImageURI(). Komponent mógłby wyświetlać proste zdjęcie użytkownik.

Dla TextView (user_name): textView.text = "imię". Tekst mógłby pokazywać nazwę.

Dla LinearLayout (np. badges_section): layout.setOnClickListener { }. Sekcja mogłaby po dotknięciu przenieść do innej strony z odznakami, gdzie appka pokazuje proste dane o postępach użytkownika.

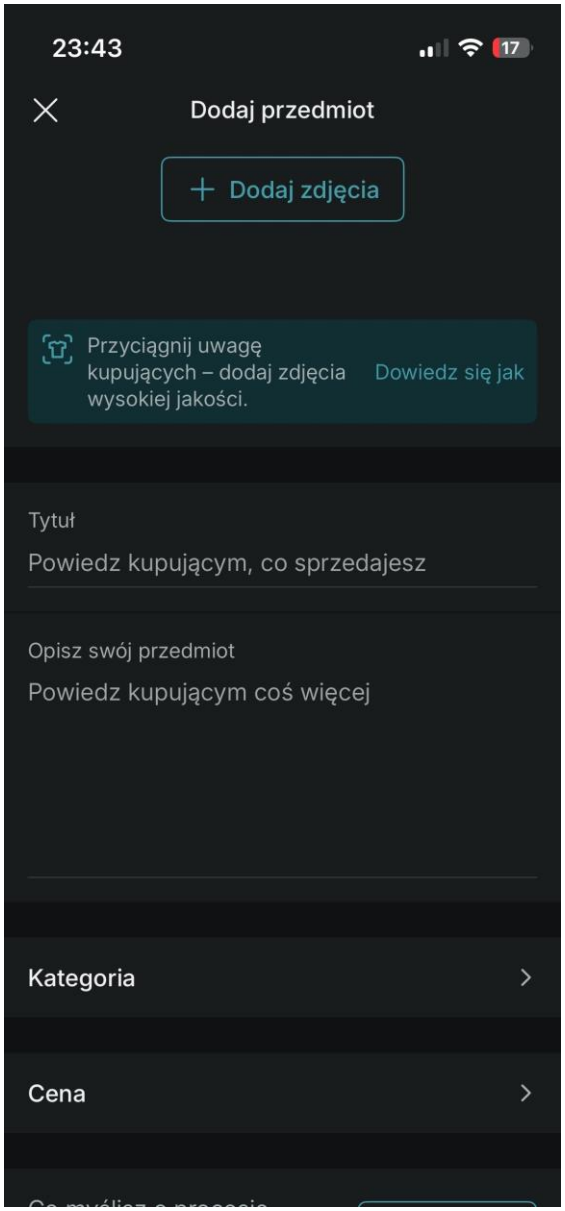
Dla innych sekcji LinearLayout: layout.setOnClickListener { }. Każda mogłaby prowadzić do swojej części appki, na przykład portfel pokazuje pieniądze, i zmieniać liczbę, gdy użytkownik coś wpłaci.

Dla Button (w sekcjach, np. portfel_button): button.setOnClickListener { }. Każdy taki przycisk mógłby po dotknięciu otwierać nowy widok, na przykład z historią transakcji w portfelu, pozwalając zobaczyć szczegóły płatności czy doładowań.



3. Widok dodawania przedmiotu

Widok formularza do tworzenia oferty: pola na tytuł, opis, kategorię, cenę, przycisk dodawania zdjęć. Jest to forma z edytowalnymi polami i przyciskami.



Lista kontrolek:

- LinearLayout (pionowy): Główny układ formularza, układający pola pionowo; obejmuje cały widok.
- Button: Przycisk do dodawania zdjęć; na górze LinearLayout.
- TextView: Etykieta dla pola tytułu; powyżej pola wejściowego.
- EditText: Pole do wpisywania tytułu; poniżej etykiety tytułu.
- TextView: Etykieta dla opisu; powyżej pola opisu.
- EditText: Pole do wpisywania opisu (wielolinijkowe); poniżej etykiety opisu.
- TextView: Etykieta dla kategorii; powyżej przycisku wyboru.
- Button: Przycisk do wyboru kategorii; poniżej etykiety kategorii.
- TextView: Etykieta dla ceny; powyżej pola ceny.
- EditText: Pole do wpisywania ceny (numeryczne); poniżej etykiety ceny.

Opis funkcjonalności w Kotlinie:

Dla Button (add_photo_button): button.setOnClickListener { }. Przycisk mógłby otwierać proste miejsce na zdjęcia z telefonu, i pokazywać wybrane jako małe obrazki, które idą z ofertą.

Dla EditText (title_edit): editText.addTextChangedListener { }. Pole mogłoby pozwalać pisać tekst, i sprawdzać, czy nie jest za długi, a wpisane słowa stawałyby się tytułem oferty dla innych.

Dla EditText (description_edit): editText.setInputType(InputType.TYPE_TEXT). Tekst mógłby być pisany w wielu liniach, opisując rzecz, i appka sprawdzałaby, czy coś jest wpisane, zanim wyśle.

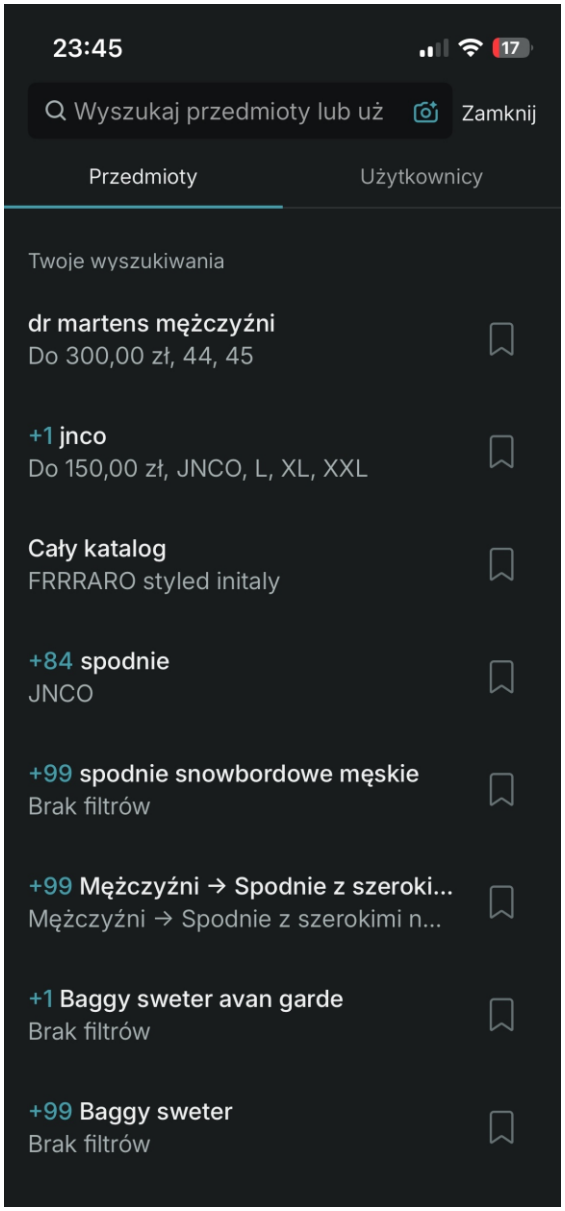
Dla Button (category_button): button.setOnClickListener { }. Przycisk mógłby pokazać prostą listę do wyboru, i zmienić swój tekst na wybraną, co pomaga appce sortować oferty.

Dla EditText (price_edit): editText.setInputType(InputType.TYPE_NUMBER). Pole mogłoby brać liczby jako cenę, i sprawdzać, czy to poprawna liczba, zanim użyje jej w ofercie.

Dla Button(submit_button): button.setOnClickListener{}. Przycisk mógłby zatiwerdzać ofertę i wyświetlać ją na profilu dla innych.

4. Widok wyszukiwania

Ekran z paskiem wyszukiwania, zakładkami (Przedmioty/Użytkownicy), listą poprzednich wyszukiwań. Użytkownik wpisuje zapytanie i widzi sugestie. Każdy element historii wyszukiwania działa jak przycisk, bo po kliknięciu odpala to wyszukiwanie, a obok paska jest przycisk do zapisywania wyszukiwania (np. do dodawania do ulubionych lub wizualnego wyszukiwania), co czyni widok pełen interaktywnych elementów do szybkiego dostępu.



Lista kontrolek:

- RelativeLayout: Główny układ widoku, pozycjonujący elementy względem siebie; obejmuje cały ekran.
- EditText: Pasek wyszukiwania z podpowiedzią; na górze RelativeLayout.
- LinearLayout (poziomy): Zakładki dla przedmiotów i użytkowników; poniżej paska wyszukiwania.
- TextView (dwie instancje): Etykiety zakładek ("Przedmioty" i "Użytkownicy"); wewnątrz LinearLayout zakładek, dzielące przestrzeń po równo.
- ListView: Lista historii wyszukiwań lub sugestii, gdzie każdy element działa jak przycisk do odpalenia wyszukiwania; poniżej zakładek, zajmująca resztę ekranu.
- Button: Przycisk do zapisywania wyszukiwania (np. Ikona zakładki książki); obok paska wyszukiwania w nagłówku.

- Button (wiele instancji, w elementach ListView): Każdy element historii to w zasadzie przycisk z tekstem zapytania, ukryty w strukturze ListView, ale klikalny; rozmieszczone w każdym wierszu listy.

Opis funkcjonalności w Kotlinie:

Dla EditText (search_edit): editText.addTextChangedListener { }. Pole mogłoby reagować na pisany tekst, pokazując proste podpowiedzi, i po naciśnięciu szukać rzeczy w aplikacji.

Dla TextView (items_tab): textView.setOnClickListener { }. Zakładki po wciśnięciu zmieniają wyszukiwanie pomiędzy przedmiotami, a użytkownikami

Dla ListView (search_history_list): Lista mogłaby wyświetlać stare wyszukiwania, a po dotknięciu wpisać je z powrotem i szukać znowu.

Dla Button (save_search_button): button.setOnClickListener { }. Przycisk mógłby po dotknięciu zapisywać bieżące wyszukiwanie do historii lub ulubionych, pozwalając szybko wrócić do niego później.

Dla Button (w elementach ListView): button.setOnClickListener { }. Każdy taki przycisk w liście mógłby po dotknięciu wpisać zapytanie do paska i automatycznie uruchomić wyszukiwanie, pokazując wyniki bez potrzeby pisania od nowa.