

**ZESPÓŁ SZKÓŁ NR. 10**

**Im. STANISŁAWA STASZICA**

**Temat: Komponenty w Pinterest**

**Patrycja Franciuk**

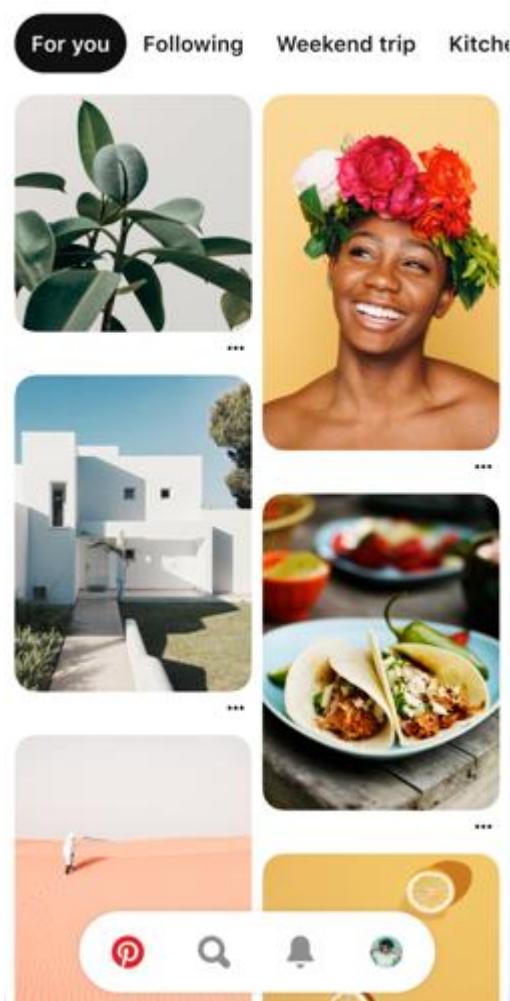
**4AP**

**Warszawa 12.01.2026r.**

## **SPIS TREŚCI**

1. Widok: Strona Główna .....	3
2. Widok: Wyszukiwanie .....	4
3. Widok: Szczegóły pinu .....	6
4. Widok: Profil użytkownika.....	8

## 1. Widok: Strona Główna



Komponent UI	Opis wyglądu elementu	Funkcja (Kotlin)
RecyclerView (Staggered Grid)	Nieregularna siatka zdjęć o różnych wysokościach.	Wykorzystanie <b>StaggeredGridLayoutManager</b> do dynamicznego dopasowania pozycji kafelków bez pustych przestrzeni.
ShapeableImageView	Obrazy Pinów z zaokrąglonymi narożnikami.	Użycie biblioteki <b>Coil</b> lub <b>Glide</b> do asynchronicznego pobierania zdjęć z URL i mapowania ich na widok.
SwipeRefreshLayout	Kontener pozwalający na	Funkcja <b>setRefreshing</b> wyzwalająca nowe zapytanie do

	odświeżenie treści gestem w dół.	ViewModelu w celu pobrania najnowszych danych.
BottomNavigationView w	Biały, zaokrąglony panel pływający na dole z ikonami: Logo, Lupa, Powiadomienia i Profil.	Logika <b>NavigationUI</b> . Kliknięcie w ikonę lupy (która jest teraz aktywna) może przewijać widok na samą górę.

## 2. Widok: Wyszukiwanie



Komponent UI	Opis wyglądu elementu	Funkcja (Kotlin)
SearchView/EditText	Zaokrąglony, jasnoszary pasek na samej górze z napisem „Search” i ikoną lupy po lewej oraz aparatu po prawej.	Obsługa <b>setOnQueryTextListener</b> . Po wpisaniu tekstu aplikacja filtruje bazę pomysłów. Ikona aparatu uruchamia intencję (Intent) otwierającą galerię (Visual Search).
RecyclerView (GridLayout)	Sekcje „Ideas you might like” oraz „Trending ideas” wyświetlane jako siatka dwóch kolumn.	Użycie <b>GridLayoutManager</b> . Dane są ładowane z listy obiektów zawierających adres URL zdjęcia oraz tytuł kategorii (np. "Breakfast recipes").
CardView + ImageView	Każdy kafelek ma mocno zaokrąglone rogi, nałożony przyciemniony obraz i biały napis na środku.	Implementacja <b>ShapeableImageView</b> z biblioteki Material Components. Logika nakłada półprzezroczystą maskę, aby biały tekst był czytelny na każdym tle.
TextView (Header)	Napisy „Ideas you might like” oraz „Trending ideas” pogrubioną czcionką.	Stylizacja <b>MaterialTextView</b> z atrybutem <b>android:textStyle="bold"</b> . Dynamiczne przypisywanie tekstu w zależności od sekcji generowanej przez serwer.

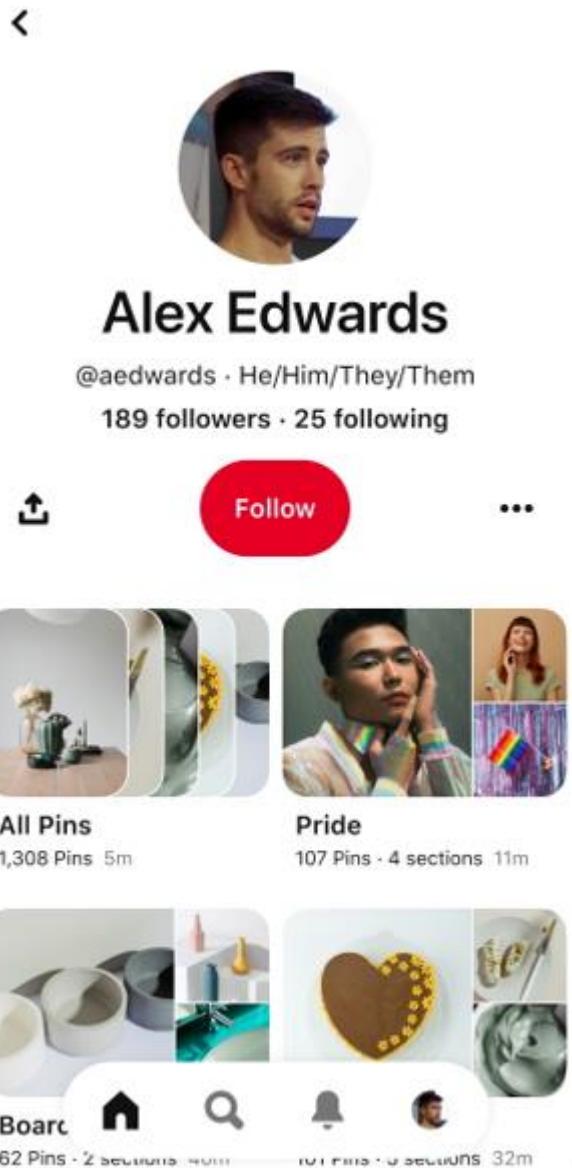
### 3. Widok: Szczegóły pinu



Komponent UI	Opis wyglądu elementu	Funkcja (Kotlin)
ImageButton (Back)	Okrągły, półprzezroczysty przycisk ze strzałką w lewym górnym rogu.	Wywołanie funkcji <b>activity.onBackPressed()</b> lub <b>findNavController().popBackStack()</b> , aby wrócić do poprzedniego ekranu.

ShapeableImageView (Main)	Główne zdjęcie wypełniające większość ekranu, z delikatnie zaokrąglonymi rogami.	Wykorzystanie biblioteki <b>Coil</b> z funkcją <b>ScaleType.CENTER_CROP</b> .
Button (Zapisz)	Duży, czerwony przycisk typu "Pill" (pigułka) z białym tekstem w prawym dolnym rogu.	Obsługa zapisu do bazy danych (Firebase/Room). Po kliknięciu wywołuje Toast lub Snackbar z informacją o zapisaniu do konkretnej tablicy.
Horizontal Stack (Social Bar)	Rząd ikon po lewej stronie (Serce, Komentarz, Udostępnij, Trzy kropki).	Logika typu <b>StateFlow</b> . Przykładowo: kliknięcie serca zmienia jego kolor i inkrementuje licznik polubień w ViewModelu.
CircleImageView (Avatar)	Mała, okrągła miniatura z literą "B" (lub zdjęciem) obok nazwy użytkownika "Barbara".	Dynamiczne generowanie tła za pomocą <b>ColorGenerator</b> (jeśli brak zdjęcia) oraz przypisanie <b>onClickListener</b> do profilu autora.
ConstraintLayout	Cały układ dolny (pod zdjęciem), gdzie elementy są ułożone względem siebie.	Użycie więzów <b>constraints</b> , aby tekst "Barbara" zawsze znajdował się po prawej stronie awatara, a przycisk "Zapisz" był przypięty do prawej krawędzi.

#### 4. Widok: Profil użytkownika



Komponent UI	Opis wyglądu elementu	Funkcja (Kotlin)
CircleImageView	Duże, okrągłe zdjęcie profilowe centralnie pod przyciskiem wstecz.	Dynamiczne ładowanie obrazu profilowego z filtrem <b>CircleCropTransformation</b> przy użyciu biblioteki Coil/Glide.
TextView (Headline)	Duża, pogrubiona nazwa użytkownika oraz mniejsze dane pod spodem	Wykorzystanie <b>String.format()</b> do łączenia danych o liczbie obserwujących (np. "189")

		<code>followers")</code> pobranych z obiektu User.
MaterialButton (Follow)	Duży, czerwony przycisk o kształcie "Pill" z napisem "Follow".	Implementacja logiki <b>toggle</b> : po kliknięciu zmiana tekstu na "Following", koloru tła i wysłanie żądania POST do API.
RecyclerView (Grid)	Sekcja tablic (All Pins, Pride) ułożona w dwóch kolumnach.	Użycie <b>GridLayoutManager()</b> . Każdy element to kafel z podglądem kilku zdjęć, co wymaga zagnieżdzonego układu lub CustomView.