
Dokumentacja Projektowa

Aplikacja Mobilna: *To-Do-List*

Autor: Bartek Domański

31.01.2026

Streszczenie

Dokumentacja opisuje proces tworzenia, architekturę oraz funkcjonalność aplikacji mobilnej **To-Do-List**. Projekt został zrealizowany w technologii Android Native (Kotlin) przy użyciu biblioteki Jetpack Compose. Celem aplikacji jest pomoc w organizacji codziennych obowiązków poprzez prostą listę zadań.

Spis treści

1	Opis aplikacji	2
1.1	Cel projektu	2
1.2	Grupa docelowa	2
2	Wybrana technologia	2
3	Architektura aplikacji	2
4	Zaimplementowane funkcjonalności	2
5	Instrukcja uruchomienia	3
6	Napołkane problemy i rozwój	3
6.1	Problemy	3
6.2	Możliwości rozwoju	3
7	Zrzuty ekranu	4

1. Opis aplikacji

1.1. Cel projektu

Celem projektu było zaprojektowanie i zaimplementowanie aplikacji mobilnej typu *To-Do List*. Aplikacja rozwiązuje problem zapominania o codziennych obowiązkach, oferując proste narzędzie do zapisywania i odhaczania zadań.

1.2. Grupa docelowa

Główną grupą docelową są studenci oraz osoby pracujące, które potrzebują szybkiego i minimalistycznego narzędzia do zarządzania czasem, bez zbędnych i skomplikowanych funkcji.

2. Wybrana technologia

Projekt zrealizowano zgodnie z wymaganiami w technologii natywnej:

- **Język programowania:** Kotlin – jest to obecnie rekomendowany język dla platformy Android.
- **Interfejs użytkownika:** Jetpack Compose – nowoczesne, deklaratywne podejście do budowania UI, zapewniające responsywność i estetykę.
- **Baza danych:** SharedPreferences + Gson – wybrano to rozwiązanie ze względu na prostotę struktury danych (płaska lista) i łatwość implementacji trwałego zapisu.

3. Architektura aplikacji

Aplikacja oparta jest na uproszczonym wzorcu architektury, w którym główna aktywność (`MainActivity`) zarządza stanem aplikacji.

- **Model:** Klasa danych `Task` reprezentująca pojedyncze zadanie (ID, nazwa, stan wykonania).
- **Widok (View):** Komponenty Composable (`HomeScreen`, `AddScreen`, `SettingsScreen`).
- **Logika:** Funkcje zarządzające stanem (`addTask`, `removeTask`) umieszczone w głównej funkcji kompozycyjnej.

4. Zaimplementowane funkcjonalności

Zgodnie z wymaganiami projektowymi, zaimplementowano następujące elementy:

1. **Wiele ekranów (Nawigacja):** Aplikacja posiada 4 ekrany: Splash Screen, Lista Zadań, Dodawanie Zadania, Ustawienia.
2. **Interakcja z użytkownikiem:** Obsługa przycisków, pól tekstowych oraz przełączników (Switch).
3. **Przechowywanie danych:** Lista zadań oraz ustawienia motywów są zapisywane w pamięci urządzenia i nie znikają po restarcie.
4. **Lista elementów:** Wyświetlanie dynamicznej listy zadań za pomocą komponentu `LazyColumn`.
5. **Estetyka (Tryb Ciemny):** Pełna obsługa Dark Mode z możliwością przełączania w ustawieniach.

5. Instrukcja uruchomienia

Aby uruchomić projekt:

1. Pobierz kod źródłowy z repozytorium GitHub.
2. Otwórz projekt w środowisku **Android Studio** (wersja Koala/Ladybug lub nowsza).
3. Poczekaj na pobranie zależności Gradle (Sync).
4. Uruchom aplikację na emulatorze (np. Pixel 7) lub fizycznym urządzeniu z Androidem 8.0+.

6. Napotkane problemy i rozwój

6.1. Problemy

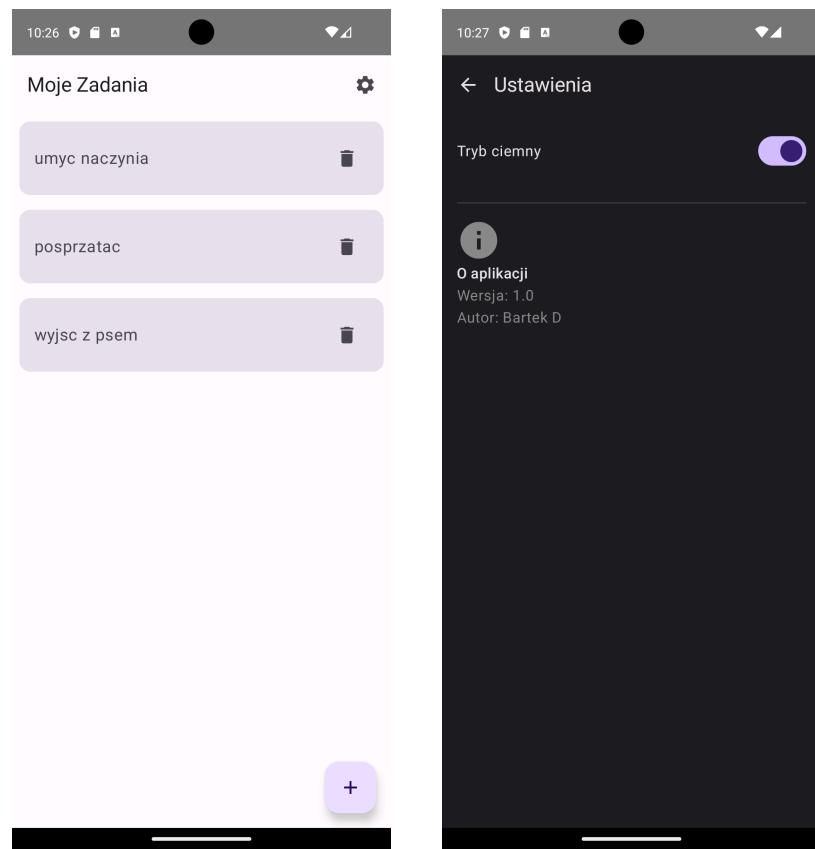
Głównym wyzwaniem była konfiguracja zapisu listy obiektów do prostego formatu klucz-wartość. Problem rozwiązano, używając biblioteki **Gson** do serializacji listy obiektów Task do formatu JSON.

6.2. Możliwości rozwoju

W przyszłości aplikację można rozbudować o:

- Powiadomienia Push przypominające o zadaniach o konkretnej godzinie.
- Podział zadań na kategorie (np. Dom, Praca, Studia).

7. Zrzuty ekranu



Rysunek 1: Ekran główny listy zadań oraz ekran ustawień.